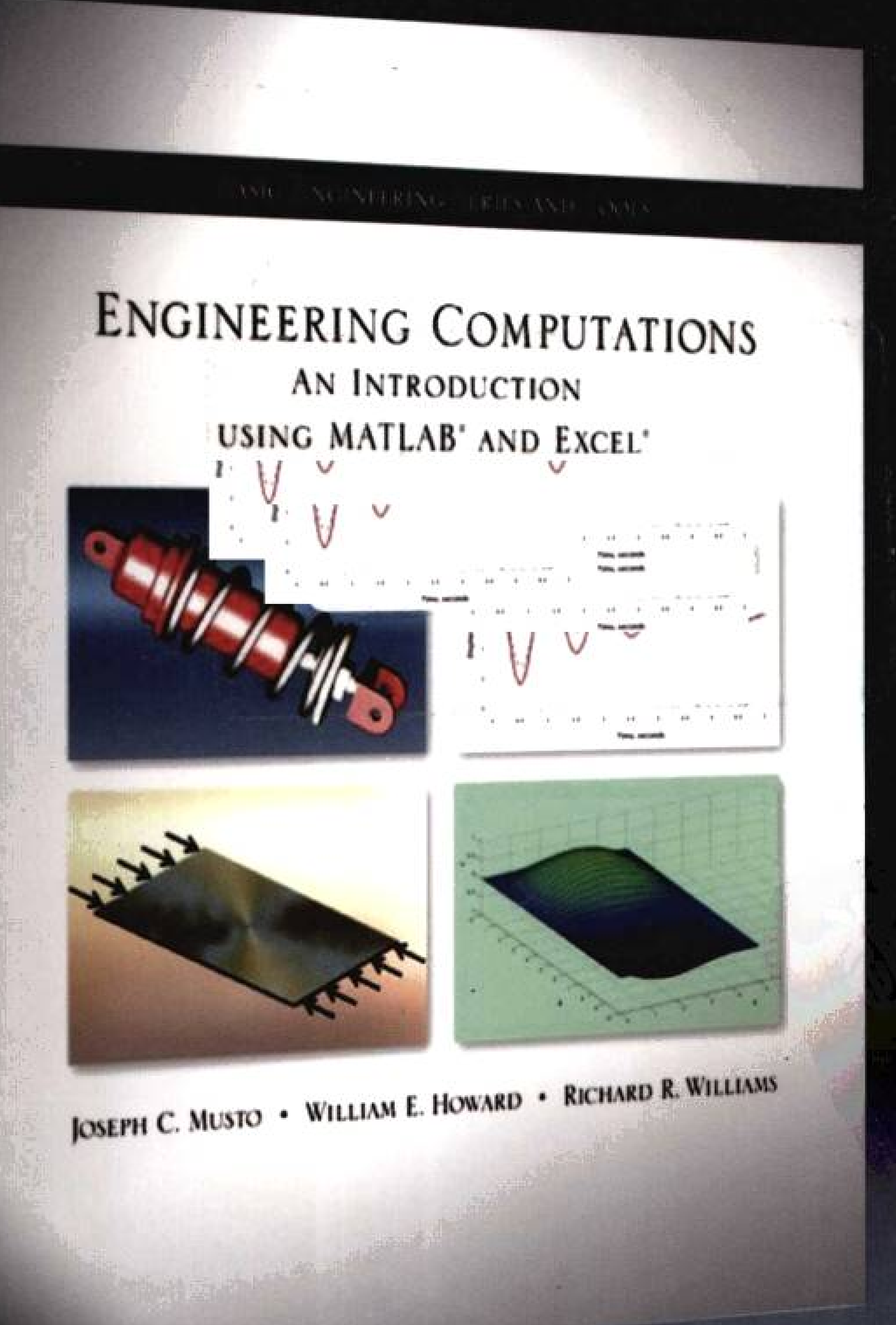
国外计算机科学经典教材

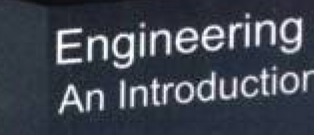
MATLAB & Excel 工程计算

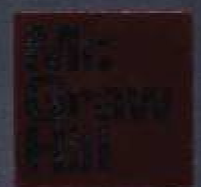
Joseph C. Musto
(美) William E. Howard 著 吴文国 林川 译
Richard R. Williams





Engineering Computations
An Introduction Using MATLAB and Excel





清华大学出版社

国外**计算机科学经典**教材

- 面向对象分析与设计（UML 2.0版）
- 数据结构与算法——C++版（第3版）
- 数据挖掘原理与应用——SQL Server 2005 数据库
- 嵌入式系统——体系结构、编程与设计
- 信息安全原理（第2版）
- C++大学教程（第3版）
- 计算机网络教程（第4版）
- MATLAB科学计算
- Linux系统管理完全手册
- 嵌入式C语言编程与Microchip PIC
- 数据库设计、应用开发和管理
- Linux系统管理完全手册
- Java大学教程（第2版）
- 卓有成效的软件项目管理
- 关系数据库和SQL编程
- 多媒体技术及应用（第7版）
- 软件测试的有效方法（第3版）
- Visual Basic.NET程序设计（第6版）
- Unix原理与应用（原书第4版）
- 编程语言：原理与范型（第2版）
- Linux管理基础教程（第4版）
- 数据仓库工具箱
- 网格计算核心技术
- 无线网络原理与应用

本书结合了当前工程计算领域里两个最重要的计算工具——MATLAB和Excel，它们是工程专业学生为解决未来工作中遇到的实际工程问题而必须掌握的计算工具。

本书不仅介绍了这两个计算工具的用法，而且由于这些计算工具不断变化，因此本书把重点放在工程计算的基本原理上，即算法设计、计算工具的选择、解决过程的文档说明、计算结果的验证和说明。

本书首先介绍工程计算的基本原理和算法的设计思想，然后介绍了Excel和MATLAB的基本用法，最后以方程求根、矩阵计算、数值积分和最优化等实例，说明工程计算的原理和思想。全书每章后面都有大量的习题，这些习题都源自工程中存在的实际问题，而非传统的数学或计算机问题。

本书特色

- 首次把MATLAB和Excel两大计算工具结合在一起
- 采用了教程模式，方便学生学习
- 以工程实例介绍工程计算的基本原理和算法思想
- 每章后面的习题都与工程的实际问题有关
- 介绍了在实际工程如何选择最合适的计算工具
- 图文结合，通俗易懂

作者简介

Joseph C. Musto博士曾就职于柯达公司和Brady公司，具有广泛的咨询经历。在伦斯勒理工学院获机械工程博士学位，在有关可靠性分析和健壮性等方面出版过一本专著和众多的技术文章。Joseph C. Musto博士还是密尔沃基工学院(the Milwaukee School of Engineering)的教授和执行院长。

William E. Howard是东卡罗来纳州立大学机械工程系副教授。从美国马凯特大学获得机械工程博士学位，主要研究领域是高级复合材料结构的设计、分析和制造。并对实体模型、快速原型技术、有限元分析等方面感兴趣。


ISBN 978-7-302-21431-1
9 787302 214311 >
定价：35.00元

信息网站：http://www.tup.com.cn
http://www.tupwk.com.cn
读者信箱：wkservice@vip.163.com
投稿信箱：bookservice@263.net

McGraw Hill Education
http://www.mheducation.com

国外计算机科学经典教材

MATLAB & Excel 工程计算

清华大学出版社

北　京

Joseph C. Musto, William E. Howard, Richard R. Williams
Engineering Computations：An Introduction Using MATLAB and Excel
EISBN: 978-007-126357-3

图书在版编目(CIP)数据

MATLAB & Excel 工程计算/(美)迈斯特(Musto J.C.)，(美)霍华德(Howard W.E.)，(美)威廉(Williams R.R.)著；吴文国，林川 译. —北京：清华大学出版社，2010.1
(国外计算机科学经典教材)
书名原文：Engineering Computations：An Introduction Using MATLAB and Excel
ISBN 978-7-302-21431-1
I. M…　II. ①迈… ②霍… ③威… ④吴… ⑤林…　III. ①计算机辅助计算—软件包，MATLAB ②电子表格系统，Excel　IV. ①TP391.75 ②TP391.13

中国版本图书馆 CIP 数据核字(2009)第 204026 号

责任编辑：王　军　于　平
装帧设计：孔祥丰
责任校对：成凤进
责任印制：王秀菊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印　刷　者：北京四季青印刷厂
装　订　者：三河市李旗庄少明装订厂
经　　销：全国新华书店
开　　本：185×260　**印　张**：19.5　**字　数**：475 千字
版　　次：2010 年 1 月第 1 版　**印　次**：2010 年 1 月第1次印刷
印　　数：1～4000
定　　价：35.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770177 转 3103　　产品编号：032699-01

出版说明

近年来，我国的高等教育特别是计算机学科教育，进行了一系列大的调整和改革，亟需一批门类齐全、具有国际先进水平的计算机经典教材，以适应我国当前计算机科学的教学需要。通过使用国外优秀的计算机科学经典教材，可以了解并吸收国际先进的教学思想和教学方法，使我国的计算机科学教育能够跟上国际计算机教育发展的步伐，从而培养出更多具有国际水准的计算机专业人才，增强我国计算机产业的核心竞争力。为此，我们从国外多家知名的出版机构 Pearson、McGraw-Hill、John Wiley & Sons、Springer、Cengage Learning 等精选、引进了这套“国外计算机科学经典教材”。

作为世界级的图书出版机构，Pearson、McGraw-Hill、John Wiley & Sons、Springer、Cengage Learning 通过与世界级的计算机教育大师携手，每年都为全球的计算机高等教育奉献大量的优秀教材。清华大学出版社和这些世界知名的出版机构长期保持着紧密友好的合作关系，这次引进的“国外计算机科学经典教材”便全是出自上述这些出版机构。同时，为了组织该套教材的出版，我们在国内聘请了一批知名的专家和教授，成立了专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动，各位委员根据国内外高等院校计算机科学及相关专业的现有课程体系，并结合各个专业的培养方向，从上述这些出版机构出版的计算机系列教材中精心挑选针对性强的题材，以保证该套教材的优秀性和领先性，避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量，我们为该套教材配备了一批经验丰富的编辑、排版、校对人员，制定了更加严格的出版流程。本套教材的译者，全部由对应专业的高校教师或拥有相关经验的 IT 专家担任。每本教材的责编在翻译伊始，就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华，在经过翻译、排版和传统的三审三校之后，我们还请编审委员或相关的专家教授对文稿进行审读，以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限，该套教材在出版过程中很可能还存在一些遗憾，欢迎广大师生来电来信批评指正。同时，也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材，共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

国外计算机科学经典教材



译者序

译者自以为比较精通 MATLAB 和 Excel 两个软件，而且曾多次给在校的大学生和校外的学员讲授过 Excel 和 MATLAB 课程，但是当我看到 Joseph C. Musto 等合著的《MATLAB & Excel 工程计算》一书时，才发现这些常用的计算工具却深藏奥秘。从头到尾浏览一遍，发出一番感慨："原来如此！"。MATLAB 和 Excel 是两个大家都比较熟悉的计算工具，与它们相关的参考书也很多，但是那些参考书只单纯介绍它们的操作和使用，缺少对它们的高级函数的深入介绍和对实际应用的讲述。而本书是从工程计算的实际应用角度出发，以这两个软件为工具，介绍工程计算的基本原理和基本方法，包括提出问题、建立模型、确定求解方法、求解过程、分析结果等内容。本书不仅"授人以鱼"，而且"授人以渔"，本书的目的是要帮助读者解决实际中遇到的工程问题，因此它的主要内容是向读者介绍解决工程计算问题的方法和步骤。本书有以下 3 个特点：

第一，理论性较强。本书以 MATLAB 和 Excel 为工具，系统地介绍了工程计算的基本概念和基本原理。这样，通过对本书的学习，读者不仅知其然，也知其所以然。

第二，理论与实践相结合。本书以工程计算的基本原理为指导，以 MATLAB 和 Excel 为工具，以实际工程问题为实例，详细地介绍了工程计算的基本过程和基本步骤。

第三，实践性强。每章都包含一两个实际工程计算实例，这些例子并不复杂，而且都是工程专业的读者所熟悉和容易接受的。读者在掌握了这些例子后，很容易举一反三，把它们应用或推广到自己的专业，而且这些例子引人入胜，能激发读者的学习兴趣。另外，每章后面都有大量的习题，以帮助读者巩固该章的内容。

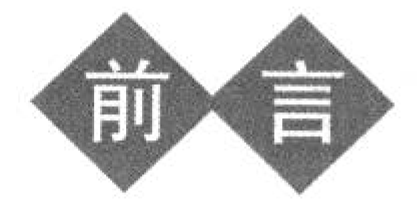
全书一共 10 章，分为两个部分。前 5 章通过实例介绍工程计算的基本概念和基本步骤，介绍 MATLAB 和 Excel 工具软件的使用和操作步骤。第 6 章~第 10 章为本书的第Ⅱ部分，以实际工程问题为例，介绍矩阵计算、方程求解、数值积分、最优化等问题。

本书的前 5 章由温州大学吴文国老师翻译，第 6 章~第 10 章由温州大学林川老师翻译。吴文国老师负责全书的统筹和校对。中国矿业大学研究生韩毅认真仔细地校对了全书。在翻译过程中还得到了多个单位的同事们的帮助，在此表示感谢。

最后，也是最重要的是，感谢清华大学出版社第五事业部。在与他们的合作过程中，我们一直非常愉快，并且从他们身上学到了许多东西。李万红老师强烈的事业心和责任感，让我非常感动。编辑们的敬业精神和认真仔细的态度时刻都在督促我在以后的翻译工作中更上一层楼。

虽然译者比较熟悉 MATLAB 和 Excel 这两个工具软件的使用，但是由于语言的差异，把原书很好地翻译成中文也确实不是一件容易的事，因此翻译中难免存在错译和误译。恳请读者批评指正，批评和建议请发送到电子邮箱 wkservice@vip.163.com，不胜感激！

吴文国，林川



作者曾给来自两个不同学院、不同学科的工程专业学生教授工程计算导论课。本书就是来自作者的实际教学经验。对于教师来说，把工程项目中用到的各种计算工具集成为一本书，肯定是一个挑战。设计计算机应用的导论课有很多目的，但通常包括以下几个方面：

- 介绍过程计算的基本原理和算法的设计思想。
- 介绍高年级课程以及毕业后的专业实践中将用到的计算工具的基本用法。
- 为学生提供解决实际问题所需要的设计方法。
- 为学生提供相关的文字说明，帮助他们选择合适的计算工具以解决当前的工程问题。
- 向学生介绍文档技术和对计算机的运算结果与实际工程问题进行验证。
- 通过向学生介绍令人感兴趣的真实问题和实际应用，从而激发学生们学习高年级课程的兴趣。

本教程选择 MATLAB 和 Excel 计算工具包，介绍上述几个概念。之所以选择这两个软件包，原因是：

- MATLAB 是许多工程问题的首选计算工具。
- MATLAB 有一个重要特性，即它既可以作为导论性的程序设计工具，也可以作为高级的计算工具。同时它既可以作为工程专业学生首次学习程序逻辑结构(循环和选择结构)的编程语言，它自带的众多数学工具和分析工具箱也可以用来解决复杂的工程问题。
- Excel 是一个被广泛使用的电子表格应用程序。几乎每个工程专业的学生在大学期间和职业生涯中都要用到 Excel。Excel 内置了许多功能强大的函数，这些函数可以应用于复杂的工程问题。
- 由于电子表格的求解方法不同于用程序设计工具开发的过程求解方法(如 MATLAB)，因此通过对这两种不同方法的比较和讨论，可以帮助学生根据求解方法的类型和复杂性选择合适的工具。

本教材的指导思想

编写本教材的指导思想是：

- 在工程专业学生的大学学习期间和以后的工作过程中，计算工具将会发生变化。虽然很有必要向他们介绍这些工具的使用，但是本教材的重点仅放在如何介绍工程计算的基本概念：算法的设计、计算工具的选择、求解过程的文档说明、结果验证和说明。
- 程序设计是工程师们的一个基本思想。虽然，那种“捷径式”的求解方法(如MATLAB中的隐含循环)和“打包式”的应用软件适合于高年级的学生和见习工程师，但是我们还必须向工程专业的学生传授结构化程序设计的基本思想，如循环结构、选择结构、数组结构等。这些基本概念独立于程序设计语言之外，是程序设计的基本模块，应该尽早向学生介绍。

根据上述指导思想，我们把本教材分为两部分。第I部分主要介绍程序设计的基本思想和电子表格的用法。具体有：

- 计算理论的简单基础。
- 数值表示方法(标量、数组和矩阵)。
- 程序的基本结构，包括算法设计和流程图表示。
- MATLAB和Excel的基本用法。
- 应用举例。这些例子包括工具选择、求解说明和结果验证。

为了指导读者使用MATLAB和Excel计算工具，上述章节提供了详细的键盘级的操作步骤。

本书的第II部分重点放在工程计算的一些典型应用。这些应用来源于实际的工程问题，具体有：

- 方程的求根
- 矩阵的运算
- 方程组的求根
- 数值积分
- 最优化

这些例子不仅要用到工程计算工具，还要用到高年级工程课程中学到的工程基本原理。在上述章节，我们不仅详细论述了这些问题的基本原理，也详细介绍了求解步骤。

为教师提供的资源

在http://www.mhhe.com/best网站有专门为教师提供的相关资源。这些资源包括每章后面的习题的答案，以及本书PowerPoint格式的电子文档。使用本教材的教师还可以与McGraw-Hill代表联系。

电子版教材

本教材还通过 CourceSmart 为老师和学生提供电子版教材。CourseSmart 是一个在线网站，在这个网站上学生可以付费访问本教材以及 McGraw-Hill 出版公司出版的其他教材的电子版。学生们只需付纸质版书籍的一半价格的费用就可以在一年内通过浏览器享用网站上的全部资源。凡是购买电子版教材的学生都可以使用 CourseSmart 网站提供的学习工具，这些工具包括全文搜索、笔记和重点说明，并且网站还提供了电子邮件工具，方便同学们之间共享学习笔记。要想进一步了解 CourseSmart 的有关内容，请与 McGraw-Hill 的销售代表联系，或者访问 http://www.CourseSmart.com 主页。

致谢

首先，我要感谢 McGraw-Hill 出版公司的支持和鼓励。在编写本书过程中，我们得到了本书的编辑 Lora　Neyens 和责任编辑 Bill Stenquist 的大力支持和指导。感谢 Fleck's Communication 为本书排版、Nicole Schlutt 为本书润稿，此外我们还得到了 MathWorks Book Program 项目的合作和支持。

本人非常重视在审稿阶段得到的反馈意见，根据这些意见才得到本书的终稿。感谢以下审稿人员仔细校对了本书的初稿，他们是：

- Ali Elkamel，滑铁卢大学
- Bill Elmore, 密西西比州立大学
- Howard Fulmer，维拉诺瓦大学
- Rrian Grady，俄克拉荷马州大学
- Mark Kerstetter，西密歇根大学
- Leo Pérez y Pérez，加利福尼亚州立大学，长滩分校
- Michael Robinso，罗斯•霍曼理工学院
- David Rockstraw，新墨西哥州立大学
- Scott Short，北伊立诺斯州立大学
- Elisa H. Barney Smith，波易斯州立大学
- J. Steven Swinnea，德克萨斯大学，奥斯汀分校
- Michael Weinstein，罗切斯特大学

东卡罗莱纳大学的学习“工程中的计算工具”这门课的学生测试了本书的最早版本，感谢这个班的同学向我提供的反馈意见。东卡罗莱纳大学的 Scott Martin 仔细阅读了本书，感谢他对很多问题的独到见解。

Joe Musto
Ed Howard
Rich Williams



目录

第Ⅰ部分　计 算 工 具

第 I 部分

计算工具



第 1 章

计算工具

引言

工程专业是专门用来解决实际问题的一类专业。它利用数学和科学的原理，解决实际工程问题，这些问题涉及建筑结构、机械、电子线路和其他各种物理系统和器件。计算机之所以成为工程师们的实用工具，是因为它具有数值分析和数据处理的能力。工程专业的毕业生，不管他们来自哪个学科，都必须精通众多的计算工具和计算软件。未来的工程专业毕业生必须能够熟练使用以下工具或软件：

- 通信工具(e-mail 和短消息)
- Internet 搜索工具
- 字处理工具(用来撰写报告和备忘录)
- 演示工具(用于音频和视频演示)
- 数据获取工具(读取和应用来自实验的数据)
- 计算工具(编程、数据分析、方程求解和绘图)

除了上述适用于所有工程专业的基本工具外，还有一些专门的计算机工具软件，它们是各自工程领域的重要组成部分。这些工具有：

- 实体模型和计算机辅助设计软件(用于机械和民用工程师)
- 电子线路仿真软件(适用于电子和计算机工程师)
- 有限元分析软件(适用于机械、民用和电子工程师)
- 高级程序设计语言(适用于计算机和软件工程师)
- 统计分析软件(适用于工业工程师)

本教材专门用来介绍如何把计算工具应用于实际工程问题中。在各工程领域里人们已经开发了各种类型的数学分析和数据处理的计算工具。虽然没有一个可称得上是“标准”

的工具可以供所有的工程师们使用，然而本书要介绍的两个软件包——MATLAB 和 Excel 却是人们熟知并且广泛使用的两个软件。我们之所以介绍这两个软件包，不仅因为它们是最受工程师们欢迎的计算机工具，还是因为它们正好代表两种不同的计算工具，即编程工具和电子表格工具。虽然这两个工具提供了截然不同的解决工程问题的方法，但是它们之间仍然存在一些相似性，特别是在数据的表示、存储和处理的方法方面。掌握这些相似性是很有必要的，因为它们提供了一种这些工具和其他计算工具都通用的程序设计语言。

在本章，读者要学习以下内容：

- 掌握解析解与算法解之间的差别。
- 掌握算法设计(algorithm development)和算法的伪代码表示。
- 掌握编程工具与电子表格工具的主要区别。
- 掌握数据存储和处理的基本术语。
- 掌握精确度与精度之间的差别。

1.1 解析解和算法解

本书中所指的计算工具允许我们自动对工程问题进行数学分析。为了更好地理解在工程问题中使用计算工具存在的优点和局限性，我们必须理解解析解和算法解两者之间的本质区别。

我们考虑一个经典的抛物运动例子。该问题在普通物理课中已经讨论过。假设从一个玩具炮架在角度为 35° 方向发射一个初速度为 10m/s 的小球，如图 1-1 所示。工程师们经常需要回答以下这些问题：炮弹可达到的最高高度、落地的位置和总的飞行时间。

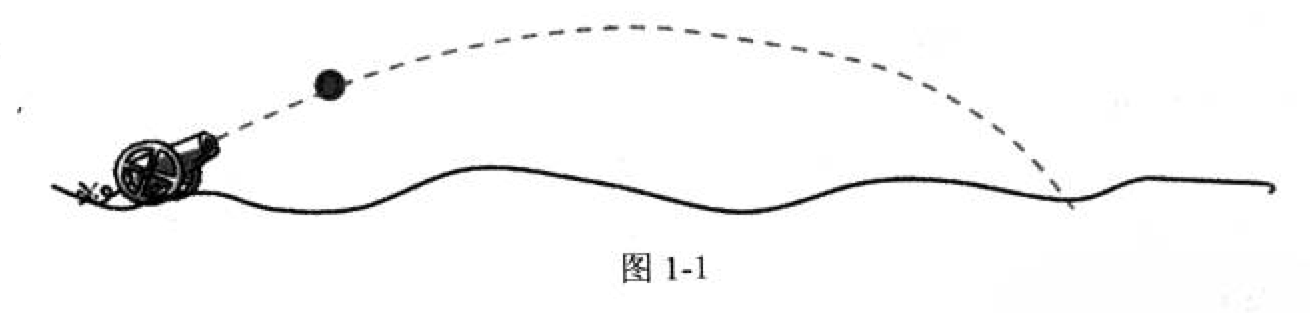


图 1-1

1.1.1 数学模型

解决该问题的第一步是建立一个数学模型，工程师利用这个模型预测炮弹的飞行规律。本例中要用到运动学的基本定律。为此，工程师们画出该系统的一个草图，如图 1-2 所示。

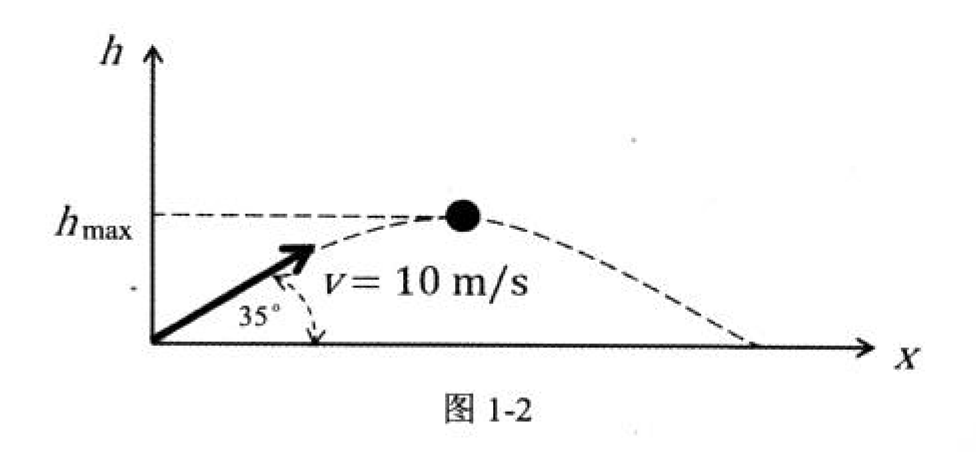


图 1-2

要建立能够描述炮弹运动规律的数学方程，我们必须做出决策——此模型要包括哪些内容。为此，我们必须在模型的准确性(是指方程预测系统行为的能力)与模型的简洁性方面做出平衡。在本例中，我们做出以下假定：

- 地面是水平的。
- 发射点在水平地面上。
- 不需要考虑风的阻力。

要做出这样的假定，或者简化假设，需要有相当强的工程判断力。工程师们要确定，在方程中考虑上述这些因素所带来的复杂性不会明显影响方程的准确度。在本例中，基于这些假设，就可以利用物理学中的基本原理，写出如下所示的两个方程，它们描述了炮弹飞行的高度和水平距离与时间的关系：

$$h(t) = vt\sin\theta - \frac{1}{2}gt^2 \tag{1.1}$$

$$x(t) = vt\cos\theta \tag{1.2}$$

这里，h 代表炮弹的高度，x 是炮弹飞行的水平距离，v 是它的初速度，θ 是发射的角度。g 是重力加速度。t 是发射后的时间(单位为 s)。有了这个模型，工程师们现在要选择一个可以求解此方程的方法。我们现在就要对比该问题的解析解与算法解。

1.1.2 解析解

解析解是精确解，它是基于对代数、微积分等数学原理的应用。在前面刚建立的模型里，可以求出它的解析解。为了求得炮弹飞行的最高高度，我们对方程(1.1)求导数：

$$\frac{\mathrm{d}h(t)}{\mathrm{d}t} = v\sin\theta - gt \tag{1.3}$$

当这个导数为零时，炮弹飞行的高度达到极限(最大值或最小值)。把这个导数设置为 0，并求 t 的值，有：

$$t = \frac{v\sin\theta}{g} \tag{1.4}$$

假如炮弹的初速度、角度和重力加速度等都已知，则可以把它们的值代入上式，经代数运算后，得到：

$$t=\frac{\left(10.0\frac{\text{m}}{\text{s}}\right)\sin(35.0^{\circ})}{\left(9.81\frac{\text{m}}{\text{s}^2}\right)} \tag{1.5}$$

其结果是 0.585s。这表示，炮弹经过 0.585s 后到达最高点。把这个值代入方程(1.1)，得到：

$$h(t=0.585\text{s})=\left(10.0\frac{\text{m}}{\text{s}}\right)(0.585\text{s})\sin(35.0^{\circ})-\frac{1}{2}\left(9.81\frac{\text{m}}{\text{s}^2}\right)(0.585\text{s})^2 \tag{1.6}$$

计算得到最高高度 h_{max}=1.68m。

为了确定整个飞行时间和水平距离，我们利用方程(1.1)，求得炮弹的高度为零时的时间：

$$\left(10.0\frac{\text{m}}{\text{s}}\right)t\sin(35.0^{\circ})-\frac{1}{2}\left(9.81\frac{\text{m}}{\text{s}^2}\right)t^2=0 \tag{1.7}$$

利用代数运算，提出因子 t，得到两个解：

$$t=0$$

和：

$$t=\frac{\left(10.0\frac{\text{m}}{\text{s}}\right)\sin(35.0^{\circ})}{\left(4.91\frac{\text{m}}{\text{s}^2}\right)}=1.17\text{s} \tag{1.8}$$

工程师们知道，对应于发射时刻 t=0，t=1.17s 是炮弹落地的时刻。把这个值代入方程(1.2)，就可以求得炮弹在 1.17s 里飞行的距离。

$$x(t=1.17\text{s})=\left(10.0\frac{\text{m}}{\text{s}}\right)(1.17\text{s})\cos(35.0^{\circ}) \tag{1.9}$$

得到的水平距离为 9.58m。

我们用标准的单位把上述结果表示为表 1-1。

表 1-1 解析解的结果

飞行的最大高度	1.68m
水平距离	9.58m
总飞行时间	1.17s

1.1.3 算法解

算法解是一个近似解。它是利用计算机程序来求解问题的。在求解算法解的过程中，工程师必须定义一系列步骤或规则，按照这些步骤或规则，得到问题的算法解。通常，算法解要利用算法原理求解工程问题，因此，虽然求得的解是近似解，但是它的好处是不需

要把复杂的数学方法应用到该问题。算法解的求解方法通过下面的实例来说明。

工程师现在面对一个方程，根据这个方程可以计算炮弹在任意时刻 t 的高度。已知炮弹在发射和落地时高度都为零，在其之间的某个时刻，它到达最高点。在飞行的前半部分，飞行高度逐渐增大，在后半部分，飞行高度逐渐减小。如果能确定这样一时刻：在这一时刻，高度不再增大，并且开始减小，那么这一时刻就是最高点的位置。我们设计了以下的算法，确定炮弹到达最高点的时刻。以下是该问题的求解步骤，即算法的伪代码：

- 步骤 1：设出发时刻 t=0，高度 h=0。
- 步骤 2：给时间t增加一个微小增量 Δt，得到新的时间t(即$t_{new}=t+\triangle t$)。
- 步骤 3：把t_{new}的值代入方程(1.1)，计算h的值，我们称这个值为h_{new}。
- 步骤 4：比较 h 与h_{new}的大小：
    - 如果$h < h_{new}$，飞行高度还在增大，表明炮弹还没有到达最高点。把t_{new}赋给t，把h_{new}赋给h，即$t=t_{new}$，$h=h_{new}$。跳转到步骤 2。
    - 如果$h > h_{new}$，则表示高度开始减小，这告诉我们，炮弹在h附近的某个位置达到最高点(或者在h与h_{new}之间，或者在前一个时间间隔)。
- 步骤 5：假设最高点出现在该间隔的开始时刻，即$h_{max}=h$。

算法解实质上是解决问题的“路线图”(road map)，它本身不是一个答案，但它是可以得到一个答案的一系列确定的步骤。该算法的主要计算部分是在步骤 2~步骤 4，在得到求解结果之前，需要反复执行步骤 2~步骤 4，但是我们事先无法确定，为得到一个算法解需要跳回到步骤 2 多少次。注意，这个算法解仅使用了算术运算。不同于解析方法，算法解不需要使用微积分或代数原理。然而，我们必须注意到，算法中使用了近似方法：只是在 t 的某些特定时刻求得高度 h，实际上，很可能最高点出现在这些特定时刻之间的某一个时刻。

如果工程师用时间步长 0.1s 重新执行上述算法解，在步骤 2~步骤 4 的反复执行过程中，每个变量的值变化如表 1-2 所示。

表 1-2 h_{max}求解过程每次循环的变量值

循环	t (s)	T_{new} (s)	h (m)	h_{new} (m)	步骤 4 判断
1	0	0.1	0	0.52	$h<h_{new}$，跳转到步骤 2
2	0.1	0.2	0.52	0.95	$h<h_{new}$，跳转到步骤 2
3	0.2	0.3	0.95	1.28	$h<h_{new}$，跳转到步骤 2
4	0.3	0.4	1.28	1.51	$h<h_{new}$，跳转到步骤 2
5	0.4	0.5	1.51	1.64	$h<h_{new}$，跳转到步骤 2
6	0.5	0.6	1.64	1.68	$h<h_{new}$，跳转到步骤 2
7	0.6	0.7	1.68	1.61	$h>h_{new}$，赋值$h_{max}=h$，算法结束

最后，该算法得到的结果是h的最大值h_{max}=1.68m。

继续讨论这个问题，现在设计一个算法，求解炮弹落在水平地面上的位置。高度值为零表示炮弹落地的位置，炮弹落地的时刻就是炮弹总的飞行时间。现在的算法如下：

- 步骤 1：设出发时刻 t=0，高度 h=0。
- 步骤 2：给时间 t 增加一个微小增量 Δt，得到新的时间 t(即 $t_{\text{new}} = t + \Delta t$)。
- 步骤 3：把 t_{new} 的值代入方程(1.1)，计算 h 的值，我们称这个值为 h_{new}。
- 步骤 4：判断 h_{new} 的值。
  - 如果 $h_{\text{new}}>0$，表示炮弹还在飞行，把 t_{new} 赋给 t，把 h_{new} 赋给 h，即 $t=t_{\text{new}}$，$h=h_{\text{new}}$。跳转到步骤 2。
  - 如果 $h_{\text{new}}<0$，则表示炮弹在 h 与 h_{new} 之间的某个位置击中地面。
- 步骤 5：代入 $t_{\text{flight}} = \frac{t + t_{\text{new}}}{2}$ 公式，求得总飞行时间的近似值。
- 步骤 6：把 t_{flight} 代入方程(1.2)求得炮弹飞行的水平距离。算法结束。

再次执行上述算法，并把时间步长设置为 0.1s。算法过程中每一步各变量值如表 1-3 所示。

表 1-3　求炮弹飞行的水平距离过程每次循环的变量值

循环	t (s)	T_{new} (s)	h_{new} (m)	步骤 4 的判断
1	0	0.1	0.52	$h_{\text{new}} > 0$,跳转到步骤 2
2	0.1	0.2	0.95	$h_{\text{new}} > 0$,跳转到步骤 2
3	0.2	0.3	1.28	$h_{\text{new}} > 0$,跳转到步骤 2
4	0.3	0.4	1.51	$h_{\text{new}} > 0$,跳转到步骤 2
5	0.4	0.5	1.64	$h_{\text{new}} > 0$,跳转到步骤 2
6	0.5	0.6	1.68	$h_{\text{new}} > 0$,跳转到步骤 2
7	0.6	0.7	1.61	$h_{\text{new}} > 0$,跳转到步骤 2
8	0.7	0.8	1.45	$h_{\text{new}} > 0$,跳转到步骤 2
9	0.8	0.9	1.19	$h_{\text{new}} > 0$,跳转到步骤 2
10	0.9	1.0	0.84	$h_{\text{new}} > 0$,跳转到步骤 2
11	1.0	1.1	0.38	$h_{\text{new}} > 0$,跳转到步骤 2
12	1.1	1.2	-0.17	$h_{\text{new}} < 0$,根据 $t_{\text{flight}} = \frac{t + t_{\text{new}}}{2}$ 计算 t_{flight}，并利用方程 1.2 计算飞行距离，算法结束

工程师把上述两个算法的结果以表 1-4 的形式报告给相关部门。

表 1-4 算法解的结果

飞行的最大高度	1.68m
飞行的水平距离	9.42m
总飞行时间	1.15s

1.1.4 解析解与算法解的比较

通过比较解析解和算法解的过程和结果，我们就不难发现两种求解方法的本质。两种方法的最重要差别是：解析解是准确解。只要采用正确的代数和微积分原理，和正确的算术运算，则解析解得到的结果精确度可以达到给定数所允许的有效位数。与此相反，算法解是近似解，不需要求得方程的精确解，但是需要计算方程在自变量 t 在某些特定值下的值。时间的这些特定值，被称为离散值，离散的结果会影响结果的准确性。根据算法解的性质，算法解只可能出现在时间 t 的某些离散值下，或两个离散值的中间值下。但是，尽管算法只能得到近似解，但是我们可以控制两个离散点之间的间隔。减小算法中的时间增量 Δt 的值，可以提高解的准确度。例如，我们把循环中的 Δt 值设置为 0.001s，则得到的结果与解析解一样。但是，为了得到这个准确值，算法中从步骤 2~步骤 4 的循环次数要增大许多倍(求 h_{max} 的循环次数从原来的 7 次变为 1680 次)。

算法解的本质决定了算法解是近似解。提高算法解的准确度的办法是减小两个离散点之间的间隔，这要增加算法的循环次数。当需要得到一个较高准确度的结果时，用手工实现算法解已被证明为不切实际的。然而，在实际极限之内，这不过是数十次、或数百次，甚至数千次重复执行算法中的某些步骤。这是因为算法解很容易用计算机工具来实现。本章前面两个算法的文字描述说明了问题求解方法的详细步骤。这些文字描述被称为伪代码。计算机程序设计语言，如本书后面将要介绍的 MATLAB 软件，很容易把独立于软件的伪代码转化为面向某个特定软件的计算机代码，关键是要提供计算机能够执行的指令。虽然计算机也可以用来执行解析解中的运算，但是算法解的执行却能最体现出计算机工具的特征。

既然解析解是准确解，算法解是近似解，那么为什么我们还使用算法解呢？在前面这个例子里，根本不需要算法解，因为我们很容易得到它的解析解。具有微积分和代数背景知识的工程师很容易得到解析解。在这种情况下，当然解析解是我们的首选。但是，在一些实际的工程专业领域里，情况并非总是如此。即使我们在大学本科阶段的学习中，也会遇到一些用数学知识无法解决的问题，甚至遇到一些根本不存在解析解的问题。正是在这些情况下，算法解可以解决那些用解析解很难求解或者不能求解的问题。如果读者曾经在可编程的图形计算器里使用过求根函数求解高次代数方程的解，那就是表示使用了由可编程图形计算器设计并实现的算法解技术。(读者是否意识到，计算器给出的结果竟然是近似值？)

算法解的另一个特点是某个问题的求解算法不是唯一的。设计求解算法需要数学推理能力和创造力的结合。计算机科学家的目标是开发出效率高、运行时短、所需内存容量小的计算算法。在本书里，我们将把重点放在实现和使用标准的求解算法上，但是在工程计

算领域里工程师将不断提出新型的、原创性的工程问题解决算法。

1.2 工程计算的方法

在本书中，我们将介绍工程计算两种不同的实现方法：编程工具和电子表格。

编程工具允许我们把伪代码算法翻译成为计算机能够识别的指令集。这些指令集就是计算机程序或计算机代码。现在有很多编程语言供实际工作中的工程师们使用。在本书中，我们利用 MATLAB 平台引入编程工具的概念。作为一个例子，我们把求炮弹最大飞行高度的伪代码算法转化为 MATLAB 指令，如下所示。在本书的第 3 章和第 4 章将详细介绍 MATLAB 程序的设计过程。

```
1   %MATLAB code for finding maximum height
2   % Definition of input variables
3   t=0; %Initial time
4   h=0; %Initial height
5   t_new=0; %New time
6   h_new=0; %New height
7   delta_t=.1; %Time step
8   v=10; %Initial speed
9   theta=35; %Launch angle
10  g=9.81; %Gravity
11  %Check for decreasing height
12  while h_new>=h;
13        h=h_new
14        t=t_new;
15        %Take one time step
16        t_new=t+delta_t;
17        %Calculate height per Eq. 1.1
18        h_new=v*t_new*sind(theta)-0.5*g*t_new^2;
19  end;
20  %Compute and output maximum height
21  h_max=h
```

在用程序实现算法时，编程工具允许我们使用逻辑判断结构。在使用计算机计算的初期，用户利用编程工具提供的指令集与计算机进行通信。尽管程序语言的外表以及与机器通信的界面已经有很大的发展，但是把算法翻译成特定软件的指令的思想仍然是计算的一个标准的模式。

电子表格工具是一个完全不同的计算模式。它代表一个很大的数据表。电子表格的用户在表格的某些单元格里输入数据，在另一些单元格里输入使用这些数据的数学方程和逻辑表达式。这种表格结构为我们提供了直观的图形方式的数据处理和计算方法。但是它与我们在编程方法中使用的逐步求解指令集有明显的差别。尽管计算的图形化方法是非常吸引人的，但是在电子表格中，与伪代码算法的直接联系有时并不十分明显。虽然市场上有很多电子表格软件，但是我们使用微软公司开发的 Excel 软件作为本书的电子表格平台。

作为一个例子，我们用一个电子表格实现前面的炮弹最大飞行高度的算法，得到的数据如图 1-3(只显示数值结果)和图 1-4(显示单元格里的计算公式)所示。

	A	B	C	D	E	F	G
1	Gravity:	9.81	m/s^2				
2	Speed:	10	m/s				
3	Angle:	35	degrees				
4	*t*	*h*	*t_new*	*h_new*		*h_max*	
5	0	0.00	0.1	0.52			
6	0.1	0.52	0.2	0.95			
7	0.2	0.95	0.3	1.28			
8	0.3	1.28	0.4	1.51			
9	0.4	1.51	0.5	1.64			
10	0.5	1.64	0.6	1.68			
11	0.6	1.68	0.7	1.61		1.68	meters

图 1-3

	A	B	C	D	E	F	G
1	Gravity:	9.81	m/s^2				
2	Speed:	10	m/s				
3	Angle:	35	degrees				
4	*t*	*h*	*t_new*	*h_new*		*h_max*	
5	0	0	=A5+0.1	=$B$2*C5*SIN($B$3*PI()/180)-0.5*$B$1*C5^2		=IF(D5<B5,B5," ")	
6	=C5	=D5	=A6+0.1	=$B$2*C6*SIN($B$3*PI()/180)-0.5*$B$1*C6^2		=IF(D6<B6,B6," ")	
7	=C6	=D6	=A7+0.1	=$B$2*C7*SIN($B$3*PI()/180)-0.5*$B$1*C7^2		=IF(D7<B7,B7," ")	
8	=C7	=D7	=A8+0.1	=$B$2*C8*SIN($B$3*PI()/180)-0.5*$B$1*C8^2		=IF(D8<B8,B8," ")	
9	=C8	=D8	=A9+0.1	=$B$2*C9*SIN($B$3*PI()/180)-0.5*$B$1*C9^2		=IF(D9<B9,B9," ")	
10	=C9	=D9	=A10+0.1	=$B$2*C10*SIN($B$3*PI()/180)-0.5*$B$1*C10^2		=IF(D10<B10,B10," ")	
11	=C10	=D10	=A11+0.1	=$B$2*C11*SIN($B$3*PI()/180)-0.5*$B$1*C11^2		=IF(D11<B11,B11," ")	meters

图 1-4

在本书的后面章节里，我们将介绍工程问题的各种求解方法。重点介绍这两种工程求解方法各自的优点和缺点。有些问题适合于编程方法，而另一些问题适合于电子表格求解方法。这就是说，本书介绍的 MATLAB 工具和 Excel 工具都是非常先进的、功能强大的计算工具，并且在某种程度上，它们都分别结合了每种方法的最好的特性。Excel 提供了一个编程界面，使得我们可以使用比较传统的计算机代码处理和填充表格里的单元格。而 MATLAB 为我们提供了矩阵编辑界面，它的外观和作用与电子表格相似。这两个工具都为我们提供了图表绘制、方程求根、最优化和其他一些常见的操作运算，用户利用这些运算可以访问工程问题求解的高级算法。本书的目标涉及两个方面。第一是要把读者培养成熟练使用这两个计算工具的高手；第二是针对某个实际工程问题，读者能够选择一个合适的解决方法。

1.3 数据表示

尽管这两种不同的计算方法存在一些差别，但是在某些方面它们还是存在一些共性。

这些共性表现在数据表示、存储和操作。在本节中，我们将介绍数据表示这个概念，并把它与本书中使用的 MATLAB 和 Excel 工具联系起来。

1.3.1 变量与函数

变量是一个量的符号表示，这个量的值不止一个。考虑以下方程：

$$y = 3x^2 + 6 \tag{1.10}$$

在该方程里，x 和 y 都是变量，它们可以取多个值。我们经常称变量为自变量或因变量。当一个变量的值由其他变量决定时，就称它为因变量。通常，在写方程时，总是把因变量写在方程等号(=)的左侧。例如，在方程(1.10)里，我们认为 y 是因变量，因为它的值由自变量 x 的值决定。当然，也可以改写上述方程，把 x 移到方程的左侧，如下所示：

$$x = \sqrt{\frac{y-6}{3}} \tag{1.11}$$

这就表示 y 就是自变量吗？不能只看方程的形式，还要分析问题的本质，才能决定哪个是自变量。我们可以肯定，x 和 y 不可能同时为自变量，因为当我们给其中一个变量设置值，就可以从方程中求得另一个变量的值。

回到前面的炮弹转变的例子。我们将炮弹的飞行高度表示成如下所示的方程：

$$h(t) = vt\sin\theta - \frac{1}{2}gt^2 \tag{1.12}$$

其中，v=初始速度；t=时间；θ=发射角；g=重力加速度。

在该方程里，有 5 个量用符号来表示。但是它们并非全部都是变量。初速度、发射角度和重力加速度是常量而非变量。在仅从方程来看这并不十分明显，但是从问题的说明来看就一目了然了。假如我们改变一下问题的描述，要求在不同角度情况下计算在 t=2s 时的高度。此时，时间 t 是常量，而发射角度是变量，但是方程的形式不变。在另一个问题里，我们需要知道炮弹在任意发射角能够达到的最大高度，在这种情况下，高度、发射角度和时间都是变量。

回到原来的问题。此时，初速度、发射角和重力加速度都是常量，因此可以认为高度是时间的函数。我们总是把 $h(t)$写在方程的左侧，如方程(1.1)所示。因此 t 是自变量，而高度 h 的值取决于 t 的值。对函数的更准确的定义是：函数是一个数学运算，输入一个值或一组值，返回单个值。这个输入值就是函数的参数。

Excel 和 MATLAB 都有很多内置函数。其中许多函数只需要一个参数。例如在 MATLAB 里 cos( )函数的输入参数是角度的弧度，函数值是它的余弦值。有些数要求多个参数。例如，Excel 有一个名为 ROUND( )的函数，它要求两个参数：需要四舍五入的数和小数位数。还有一些函数的参数个数是可变的。在 Excel 里，AVERAGE( )是一个求平均值的函数。在 MATLAB 里，有很多函数使用数组或矩阵作为参数，我们将在后面章节里讨论这个问题。(注意，本书约定，MATLAB 的函数用斜体字表示，Excel 函数用大写字母表示。)

1.3.2 标题与数组

在第 1.1 节的算法解例子里，我们根据自变量 t(时间)的值进行计算，时间 t 每增加一个 Δt，就需要计算炮弹飞行高度一次。对此进行反复计算，直到高度开始减小为止，这表示炮弹已经到达最大高度。选择时间的增量为 0.1s，我们再次把分析结果列在表 1-5 里。

表 1-5 炮弹飞行的算法结果

循环	T (s)	T_{new} (s)	h (m)	h_{new} (m)
1	0	0.1	0	0.52
2	0.1	0.2	0.52	0.95
3	0.2	0.3	0.95	1.28
4	0.3	0.4	1.28	1.51
5	0.4	0.5	1.51	1.64
6	0.5	0.6	1.64	1.68
7	0.6	0.7	1.68	1.61

在电子表格求解方法中，我们需要对电子表格的单元格进行运算。其结果与表 1-5 相似。在电子表格里，单元格里的一个数值被称为一个标量(scalar)。但是在MATLAB等编程语言里，采用另外一种不同的格式表示运算结果。我们在算法中可以使用t、t_{new}、h和h_{new}变量。在每次循环中，要覆盖这些变量的前一次循环的值。这样做，就是把每个变量当作一个标量，允许每个变量只有一个值。但是如果希望保留每次循环的结果，我们要用什么方法？有时我们还要绘制高度随时间的变化图，为此需要把这些运算结果保存在内存里，但是我们又不可能给每个数值一个唯一的变量名(例如，用$t1$、$t2$、$t3$……表示每个时间值)。如果这样的话，需要对这个问题进行顺序计算，而不能用循环进行计算。实际上，我们可以用数组保存运算结果。一个数组是一个可以表示多个值的变量。例如，时间t可以是一个保存 7 个值的数组。数组的某个值要通过索引引用。索引就是一个整数，它表示某个值在数组里的位置。我们不妨把索引看成是地址。以这里的时间t为例，它有 7 个地址，分别用 1~7 表示。在每个地址都保存时间t的一个值，如表 1-6 所示。

表 1-6 数组 t 的结构

索引	1	2	3	4	5	6	7
时间(s)	0	0.1	0.2	0.3	0.4	0.5	0.6

在数组名后面的括号里使用索引值表示(或者数组名的下标表示)数组的某个元素。例如，$t(5)=0.4$s，$t_3=0.2$s。

还要注意，数组的索引必须是整数，而且它是从 1 开始的，之后逐个加 1。刚入门的程序员最容易犯的错误有：

- 总喜欢用 0 作为数组的索引。在我们的例子里，时间的第一个值是 0，因此我们总是情不自禁地用 $t(0)=0$ 表示第一个元素。在 MATLAB 里这会引起以下的错误：

```
>> t(0) = 0
??? Subscript indices must either be real positive
integers or logicals.
```

- 总喜欢使用非整数索引。例如，语句“$t(0.1)=0.1$”也会出现同上面一样的错误信息。
- 不是用加 1 的递增方法访问每个元素。例如，假设在一个实验中，每隔 10s 记录温度的值。假设第一个温度值为 100ºC，然后就写出这样的语句 $T(10)=100$。这样做虽然不会造成错误，但是这样会得到一个从 $T(1)\sim T(9)$都为 0 的数组，如下所示。

```
>> T(10) = 100
T =
0 0 0 0 0 0 0 0 0 100
```

出现上述 3 种错误的原因是我们混淆了自变量与索引的关系。记住，索引仅是数组里表示地址的值，它本身不是变量。

上面曾提到数组 t 是一维数组。即只需要一个索引整数就可以确定数组中的某个值。除了一维数组外，还有二维数组。例如在表 1-5 里，每次循环都有两个时间值：t 和 t_{new}。与其把这些值保存在两个不同的一维数组里，不如把它们保存在一个二维数里。我们规定第一个索引的值可以取 1 和 2，分别表示 t 和 t_{new}，第二个索引表示循环的次数，则这 14 个时间值都可以保存在一个数组里。例如，根据这种方法，$t(1,5)=0.4, t(2,5)=0.5$。

1.3.3　矩阵与矢量

一维或二维数组经常被称为矩阵。矩阵不仅是一种高效存储数据的方法，更重要的是，它还可以直接进行许多数学运算。事实上，MATLAB 这个名字就是代表 Matrix(矩阵)和 Laboratory(实验室)。最初设计 MATLAB 这个程序的目的就是进行矩阵运算。在第 7 章和第 8 章里，我们将学习简单的矩阵数学运算，并且利用矩阵方法求解方程组。

一个矩阵的大小由它的行数和列数确定。例如，下面这个矩阵是一个 3×2 矩阵(读作 3 行 2 列矩阵)。它有 3 行 2 列元素：

$$\begin{bmatrix} 3 & 2 \\ 7 & -5 \\ 6 & 12 \end{bmatrix}$$

在 MATLAB 里，我们常称一维数组为矢量。如果一组数排列成单行形式，我们称它为行矢量。如果一组数排列单列的形式，我们称它为列矢量。因此一维数组也被称为列矩阵或行矩阵。

必须提醒读者，在物理学和工程力学中，矢量还有另一个定义，即矢量(a vector quantity)

是指一个用大小和方向来定义的量。例如，速度是一个矢量。要准确定义一个速度，不仅需要确定它的大小(速率)，还需要确定它的方向。定义矢量的一个方法是定义它在 x、y 和 z 三个方向上的分量。当然这三个分量可以用一维数组来表示，这正好符合 MATLAB 的矢量定义。由于这两个不同的定义可能会给我们带来混淆，因此在本书中，当我们需要表示一维数组时，就不会使用矢量(vector)这个词，而是用一个更加普通的术语——数组来表示多值变量，用矩阵(matrix)表示需要进行矩阵数学运算的一维或二维数组。

在 Excel 里，我们也可以表示和操作数组。可将输入到电子表格单元里的数据看成矩阵，可以把矩阵运算应用到这些单元格上。在 Excel 里，这些运算使用预先编制好、专门为矩阵运算而开发的函数来实现。Excel 不同于 MATLAB，后者是专门为矩阵运算而开发的。在 MATLAB 里，矩阵和标量的运算使用相同的数学符号，而在 Excel 里矩阵运算需要特殊符号。在本书的第 7 章将介绍这些运算方法。

1.3.4　准确度与精度

准确度与精度经常互换使用，但是在计算应用程序中它们还是有不同的含义。准确度是指某个计算值与实际值的接近程度，它本身是一个模型的函数。例如，在前面求炮弹最大发射高度的例子里，所取的时间步数越多，求得的结果越接近“准确的”解析解。此外还要注意这个模型里的几个假设，例如，我们忽略了风的阻力的影响。这个假设也会影响解的准确度。

解的精度取决于输入值的正确性和这些数据在计算机里的存放形式。例如，在炮弹发射这个例子里，发射角是 35°，但是这个角度值准确吗？这要取决于炮弹的发架的设置和角度的测量。这个角度值可能是精确到度，或精确到十分之一度或精确到 5°。

在科学领域里，通常输入变量的测量值的精度是已知的，计算结果的精度取决于输入值的有效位数。对于带小数点的数，它的有效位数定义为第一位非零数字与最后一位之间的位数。考虑下面的例子：

1214.55　　6 位有效数字
1214.5513　8 位有效数字
0.00012　　2 位有效数字
10.00012　 7 位有效数字

在进行运算时，运算结果的精度由输入值的最小精度决定。对于加法和减法运算，这意味着，结果中小数点右侧的位数必须等于输入值中小数点右侧的最少位数。例如：

$$6.778 + 3.5 = 10.3$$

$$10.0 - 0.0012 = 10.0$$

对于乘法和除法运算，运算结果的有效位数必须等于输入值的有效位数的最小值。例如：

$$(7.553)\ (5.52) = 41.7$$

$$1.0 / 4.5567 = 0.90$$

有些量是准确值。例如，1ft 正好等于 12in。因此，如果我们要把 11.556in 转换为 ft，则结果是：

$$(11.556\text{in})\left(\frac{1\text{ft}}{12\text{in}}\right)=0.96300\text{ft}$$

在这种情况下，$\left(\frac{1\text{ft}}{12\text{in}}\right)$这个值有无穷位有效数字。

不带小数点的数的精度一般都是无法确定的。例如，我们前面讨论过，炮弹的发射角度 35°的精度就是如此。工程问题经常出现这种情况，即总是存在一些输入量的精度无法确定。因此，前面介绍的与精度有关的运算规则不能使用，还必须说明有效数字的位数。许多工程教材建议为最终结果保留 3 位有效数字(有的教材建议如果第一位有效数字是 1，要保留 4 位有效数字)。用手工进行运算时，中间结果的有效数字位数应该保留比最终结果多几位有效位数字。例如，如果把 sin(35°)的值舍入为 0.57，就不可以在最终的结果里保留多于 2 位的有效数字。

用计算机求解时，不会对中间结果进行舍入操作，因此最终结果的精度通常取决于输入值的精度。这里必须使用“通常”这个词，因为在有些情况下，如运算中同时用到很大和很小的数，用计算机计算也会产生累积误差。例如，当我们利用一种名为有限元分析方法的计算技术分析机械结构时，需要求解数千个模拟方程，如果在这些方程中的数值相差几个数量级，则求解算法的程序必须想办法使计算误差最小化。就工程专业的学生和正在实习的工程师们经常遇到的问题而言，这不会成为一个问题。Excel 和 MATLAB 里的数值的精度到底是多少呢？Excel 保留 15 位有效数字。而在默认情况下，MATLAB 以双精度形式存储数值，它的精度也是 15 位。这里的“双精度”这个术语是指存储这样一个数需要用两个字节的计算机内存存储，而单精度数是指存储这样的一个数需要占用一个字节的计算机内存。在计算机的早期发展阶段，内存空间非常有限，因此只有为了确保运算精度时才使用双精度数值。使用双精度数也会增加计算时间。今天，由于计算机的硬件价格非常低，而且 CPU 运算速度非常快，因此，几乎没有必要使用单精度数。尽管如此，MATLAB 还是支持单精度数。因此当我们需要处理超大规模数据集时，可以使用单精度数。

关于准确度和精度，有一点需要特别引起读者的注意。许多学生在用手工计算时，很自然会把运算结果舍入到一定精度，但是当他们用计算机程序求解一个工程问题时，总是在计算结果的报告里，包括了显示在计算机屏幕上的全部位数。我们这样做就是把计算机求解过程看成是一个黑盒，根本没有考虑到在输入与输出之间的运算过程。在计算机求解结果的报告中，如果我们使用合适的有效位数，就会给他人这样的印象：我们了解与该问题有关的一些假设和近似条件。工程专业的学生和实习工程师们必须正确理解计算结果，引用计算结果时要使用合理的有效位数，不要照搬计算机的输出结果。

1.4　习题

1. 说明解析解与算法解之间的区别。
2. 设计一个伪代码算法，求函数 $f(x)=3x^2-12.4x+3$ 与 x 轴的两个交点。
3. 考虑第 1.1.1 节中使用的炮弹模型。

(a) 利用本节中的运动方程，假设发射速度为 10.0m/s，设计一个伪代码算法解，当发

射角至少为多大时，炮弹发射的最大高度可达 2.5m?

(b) 假设角度离散值的间隔为 5°，用手工方法执行这个算法解，用表格的形式输出每一步的执行结果。

(c) 求出该问题的解析解，并与算法解的结果进行比较。

4. 考虑第 1.1.1 节中介绍的炮弹飞行问题。要求炮弹飞越在发射点的正前方 8m 处的一堵 5m 高的墙，求出发射速度和发射角度的关系。假设炮弹的最大发射速度已知。设计一个求解该问题的算法解的伪代码。

5. 设计一个算法的伪代码，决定 10 个数当中有多少个是偶数。

6. 一位工程师需要测量若干钢筋的直径和长度，得到的结果如表 1-7 所示。计算它的截面积和体积，并用正确的有效数字汇报测量和计算结果。

表 1-7

钢　筋	直　径	长　度
A	0.125in	12.80in
B	0.13in	1.1ft
C	0.1250in	1.1ft
D	0.01250in	1.067ft
E	0.38in	8.11in
F	0.3750in	8.110in
G	0.3750in	8.1in
H	1.2in	5.29in
I	1.20in	5.290in
J	1.200in	5.3in

7. 利用计算机搜索并记录钢铁的密度，用这个密度值，计算习题 6 中的每个钢筋的质量。并用正确的有效数字输出计算结果。

8. 找到并阅读著名的工程历史学家 Henry Petroski 的文章“Failed Promises”，该文章发表在 1994 年 1 月~2 月的《美国科学》杂志上。写一个摘要，介绍该文章里与工程计算领域有关的内容。

9. 1969 年 7 月，美国实现了把人送到月球的宏伟目标。考虑到当时的计算机硬件和软件条件，这确实是一个奇迹。阅读美国国家航空和宇宙航行局(NASA)历史处编写的阿波罗飞行杂志(http://history.nasa.gov/afj)上 Phill Parker 的文章“the Apollo On-board Computers”(阿波罗飞船上的计算机)。写一篇摘要，对比阿波罗工程使用的计算工具与今天可以使用的工具。

第 2 章

Excel 基础

引言

Excel 计算工具常用于工程问题和非工程问题的数据分析和数据显示。它广泛应用于工业企业、商业单位和学术研究机构。Excel 以电子表格的形式实现简单的数据处理、数据分析和排序，并且用图表的形式显示运算结果。此外，它内置了许多功能强大的计算和分析工具，利用这些工具可以解决许多工程问题以及统计和金融领域里的其他相关问题。我们把自大多数工程分析仪器的数据导入到 Excel 电子表格里，然后对它们进行分析和处理。Excel 的处理结果(通常是图表和表格)很容易导出到微软的 Word 字处理软件和 PowerPoint 幻灯制作软件里，生成报告和演示文档。

本章将要学习以下内容：

- 学习 Excel 电子表格工作环境的相关术语。
- 学习如何在 Excel 电子表格里输入数据和公式。
- 学习如何用格式设置改善 Excel 电子表格的外观和功能。
- 学习如何用 IF-ELSE 命令实现逻辑判断。
- 学习如何使用查找表。
- 学习如何用 Excel 实现线性插值。

2.1 Excel 界面

单击如图 2-1 所示的 Excel 图标，启动 Excel 程序。在 Excel 程序启动后，出现一个新建的工作簿，就是由多个工作表组成的电子表格，如图 2-2 所示。



图 2-1

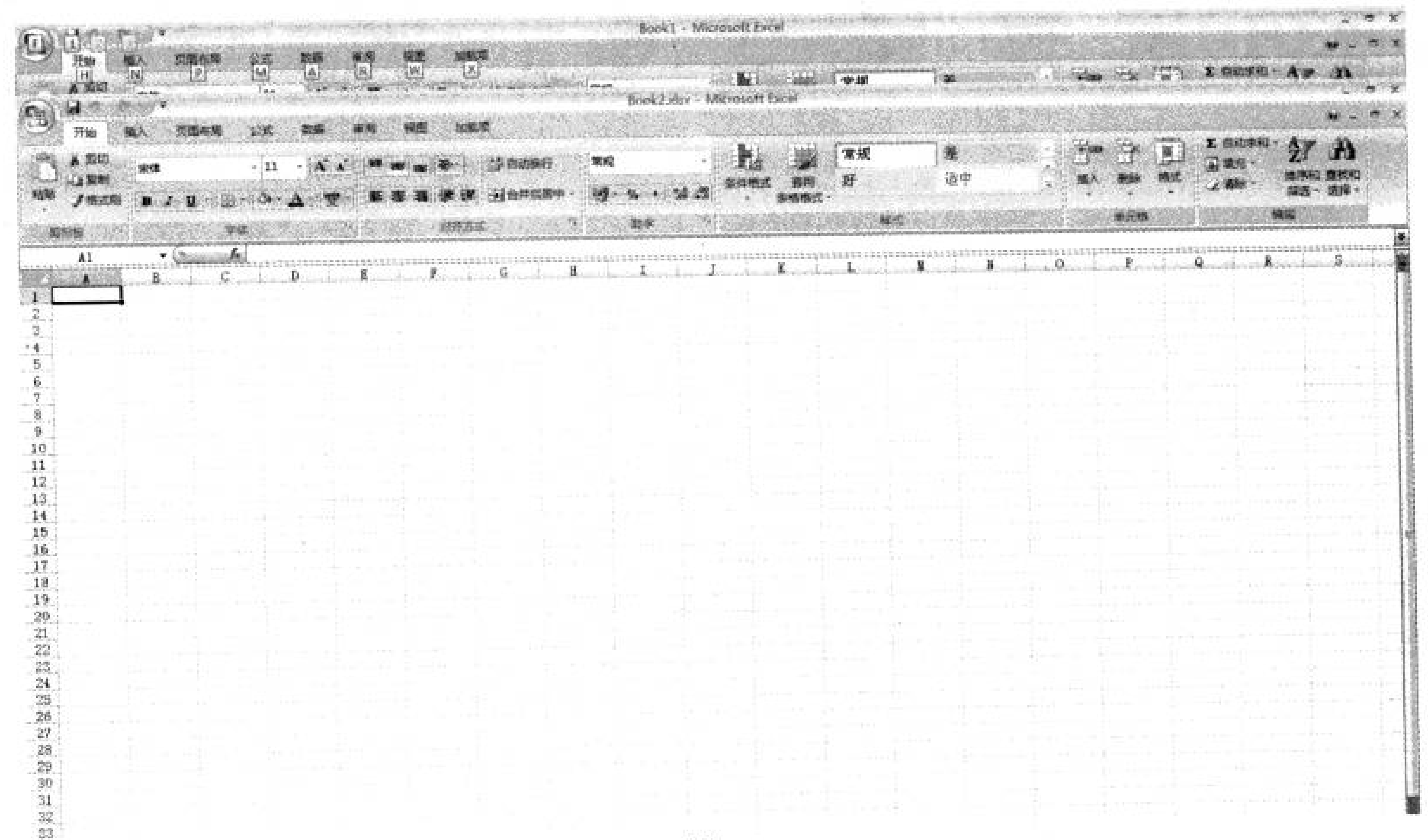


图 2-2

注意到电子表格的顶部，这里有一些工具和窗口，如图 2-3 所示。

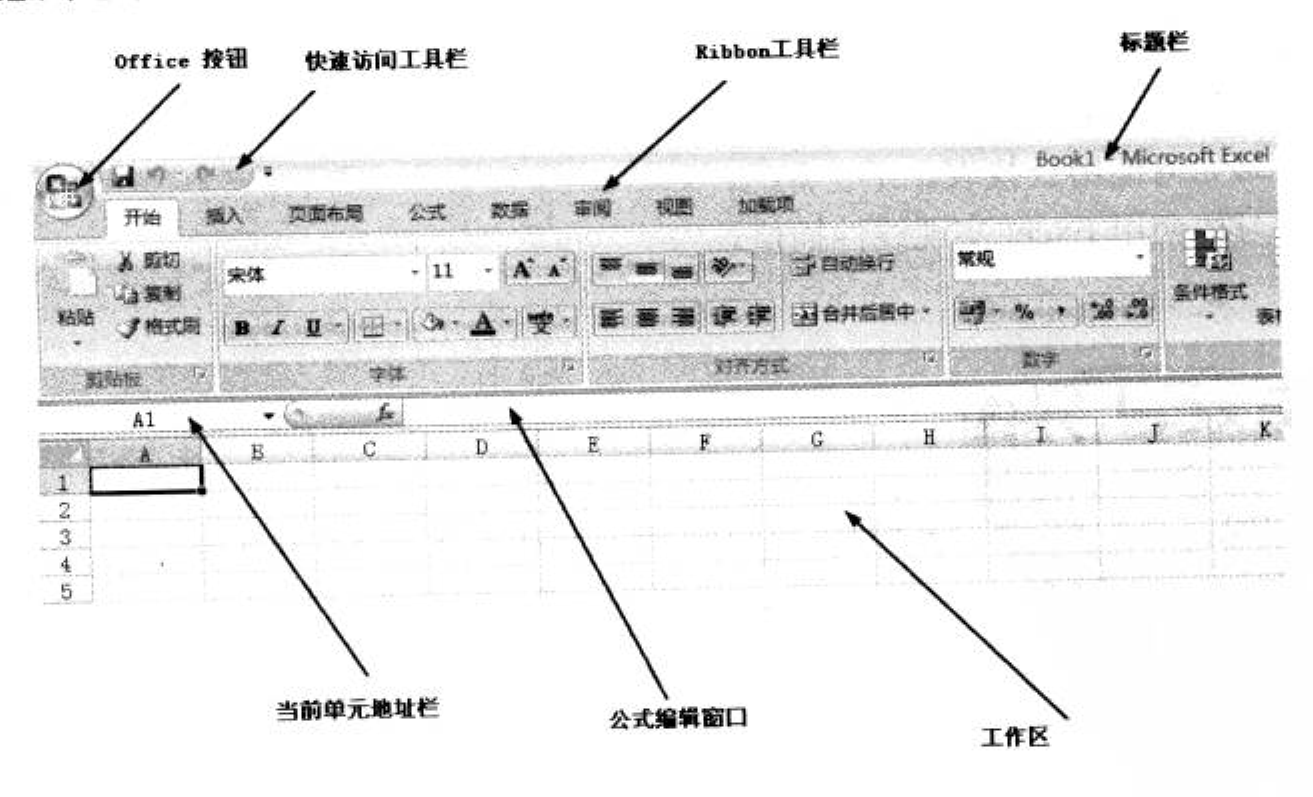


图 2-3

- Office 按钮：所有的 Microsoft Office 2007 的产品都有这个按钮。单击 Excel 中的这个 Office 按钮，会弹出一个菜单。利用这个菜单，我们可以执行建立、打开、保存、打印和用电子邮件发送一个电子表格等操作。Office 按钮还有一个链接菜单项，通过它可以进行个性化设置，或给电子表格添加其他功能。
- 快速访问工具栏：通过这个工具栏，我们可以新建或打开一个电子表格，保存、打印或邮寄一个电子表格。它还有撤消和恢复两个工具。单击快速访问工具栏右侧的向下箭头，可以根据自己的习惯进行设置。
- 标题栏：Excel 电子表格的标题栏出现在此窗口里。当同时处理多个电子表格时，标题工具会给我们带来许多便利。
- Ribbon 工具栏：这是 Excel 最重要的工具栏。它提供了建立和编辑电子表格所需要的绝大多数命令并将命令根据它们的逻辑关系进行排列。最常使用的命令都在【开始】选项卡里，出现在 Ribbon 工具栏里的工具取决于当前正在进行的操作。在每个选项卡里的命令都会根据功能进行分组，并显示最常用的命令。例如，在【开始】选项卡里，【字体】工具允许我们快速改变字体类型、大小、外观和填充颜色等。本章后面的教程将帮助读者熟悉 Ribbon 工具栏里的许多工具。在 Ribbon 工具栏的右上角还有一个形状像问号的【Excel 帮助】图标。单击该图标，打开一个帮助窗口，在这个窗口里，可以按照关键词进行搜索。
- 工作区：这是 Excel 电子表格的核心，电子表格的数据和公式都将出现在这里。工作区被分割为许多单元格，每个单元格通过它的行号和列号进行访问。例如，最左上角的单元格是在列 A 和行 1 的交叉位置，因此它的地址就是 A1。必须先“激活”单元格，才可以输入数据或编辑数据。方法是单击单元格。被激活的单元格四周有一个黑色边框，它的行和列被突出显示。
- 被激活单元格的地址窗口：此窗口显示被激活单元格或单元格组的地址。当需要选择或编辑大块数据时，这个窗口很有用。
- 公式窗口：此窗口显示公式或在被激活单元格里输入的数据。在这个窗口输入和编辑单元格里的公式和数据。

在工作区的底部还有几个按钮，如图 2-4 所示。

- 【工作表选项卡】：通过这些选项卡，我们可以选择电子表格或工作簿的任意一个工作表。每个工作表可以包含独立的数据或公式，也可以包含到其他表格的链接。右击工作表的名字，可以移动、复制、删除、隐藏、显示、插入和重命名工作表。当工作表很多、选项卡的内容不能全部显示时，利用左侧的箭头可以移动到其他工作表。此选项卡右侧的图标，可以用来插入新的工作表。
- 状态栏：状态栏显示工作表的信息、单元格个数、权限状态和宏记录状态。用鼠标右击状态栏或从菜单，可以控制状态栏显示的信息。
- 视图工具栏：利用该工具栏我们可以改变工作簿窗口的外观。当我们把【正常模式】切换到【页面布局】时，就可以看到电子表格的打印效果，而且还照样可以建立或编辑电子表格。当我们用电子表格建立报表时，这个功能非常有用。【分页预览工具】可以用来设置分页位置。视图工具也可以通过【视图】选项卡里的 Ribbon 工具栏访问。

- 【滑块】：利用这个控件，我们可以把工作簿的窗口放大到400%，也可以缩小到10%。当我们使用便携式计算机时，这个工具就会非常有用。

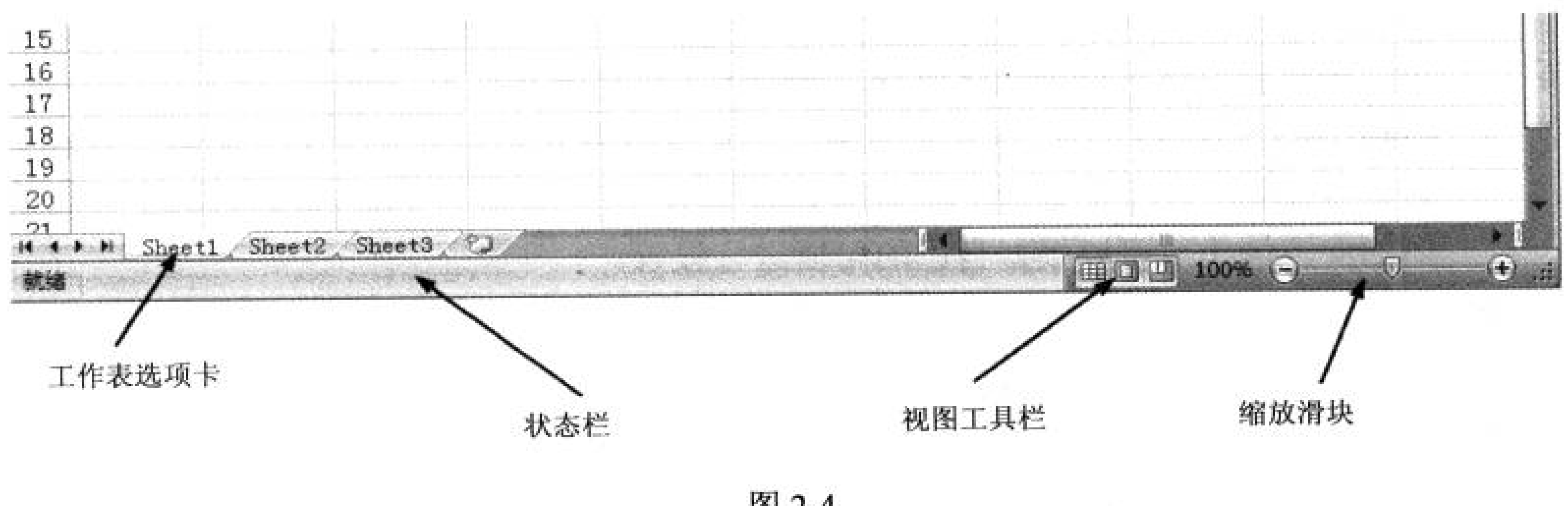


图 2-4

2.2 教程：Excel 的数据输入与格式设置

现在我们就学习如何在 Excel 电子表格里输入数据以及如何对数据进行格式设置，使得数据看起来更美观、更容易阅读。在本教程里，我们将建立一个电子表格，表格里列出各种飞行器的飞行速度记录和创立记录的日期。

首先我们在 A1 到 D1 单元格里(以后就用 A1:D1 表示)输入每一列的标题。单击 A1 单元格，输入 Aircraft Class。按同样方法，在 B1:D1 单元格里分别输入 Record Setter，Date，Speed。读者肯定会注意到，单元格宽度不够，不能显示全部内容。解决的办法就是单击两个列标号之间的竖线，就可以把列调整为合适的宽度，如图 2-5 所示。

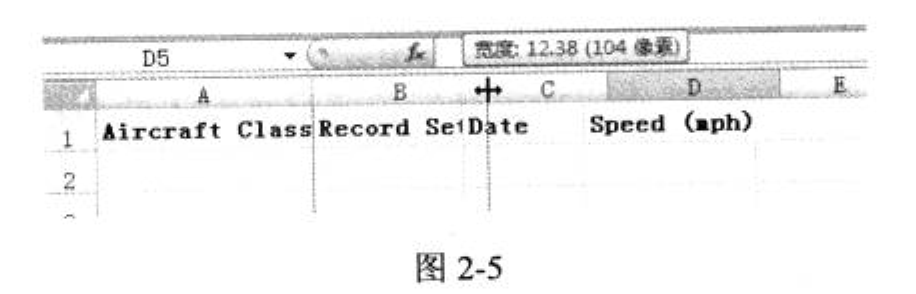


图 2-5

接着在 A2:A8 单元里分别输入 Autogyro，Rotorcraft，Biplane，Piston Powered，Turboprop Powered，Jet Powered，Rocket Powered 八种机型。

现在在 B2:B8 单元里输入创造记录的飞行器名字。依次输入：WA-116F，Westland Lynx，Fiat CR42B，Grumman F8F Bearcat，Tupolev Tu-114，Lockheed SR-71A，North American X-15A-2。然后在 C2:C8 单元里输入以下日期 9/18/1986，8/11/1986，1941，8/21/1989，4/9/1960，7/28/1976，10/3/1967。

最后，在 D2:D8 单元格里输入速度：120.5，249.1，323，528.33，545.07，2193.16，4520。

至此，我们建立了一个如图 2-6 所示的表格。它的可读性并不好，而且它当前使用的格式也不好看。

	A	B	C	D
1	Aircraft Class	Record Setter	Year	Speed (mph)
2	Autogyro	WA-116F	9/18/1986	120.5
3	Rotorcraft	Westland Lynx	8/11/1986	249.1
4	Biplane	Fiat CR42B	1941	323
5	Piston Powere	Grumman F8F	8/21/1989	528.33
6	Turboprop Pov	Tupolev Tu-11	4/9/1960	545.07
7	Jet Powered	Lockheed SR-7	7/28/1976	2193.16
8	Rocket Powere	North America	10/3/1967	4520

图 2-6

首先，我们需要调整列的宽度，下面我们使用一个与前面不同的方法。把光标移动到列标头里 A 列和 B 列的分隔线位置，双击鼠标，就会发现 Excel 自动把列宽调整为最合适状态，正好显示该列的内容。重复此操作，把 B:D 列的宽度都调整设置为最合适状态。

现在要修改表格的表头的外表。选择 A1:D1 单元，方法是单击 A1 单元，拖动鼠标到 D1 位置。然后，单击 Ribbon 选项卡中字体组里的【粗体】按钮，如图 2-7 所示。

选择 A1:D8 单元格，再单击 Ribbon 选项卡里对齐组的【居中】工具，使 A1:D8，这些单元格里的数据处于单元格的中央，如图 2-8 所示。



图 2-7

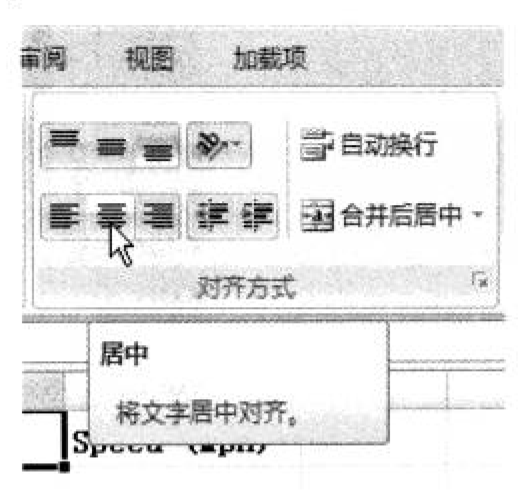


图 2-8

接着我们给表格加上边框。设置边框的操作步骤的顺序非常重要，因为某一步操作可能会撤消前一步的操作结果。首先，选择 A2:D8，给整个数据(表格列标题除外)加上一个外边框。然后，从【边框】的下拉菜单里选择【所有框线】命令。边框工具位于 Ribbon 选项卡的字体工具组里，如图 2-9 所示。

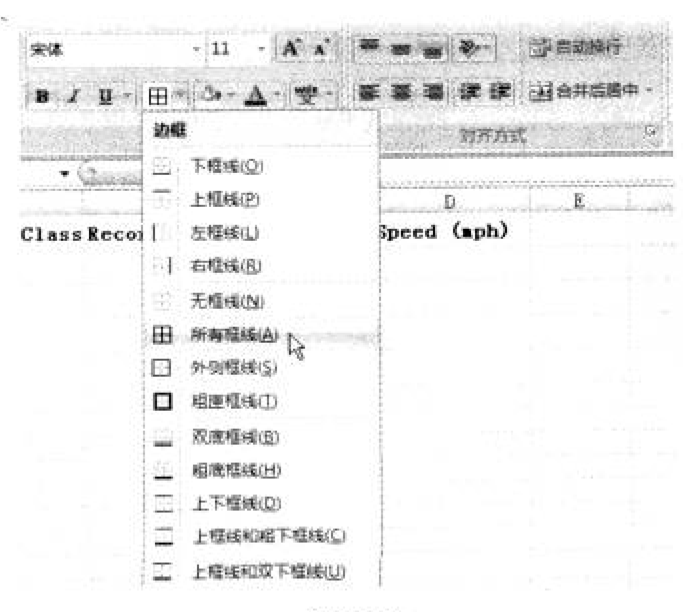


图 2-9

保持 A2:D8 单元还处于选中状态，选择【粗匣线框】命令，把选中的单元格的外层边框设置为粗线框。

接着，我们把粗线线型应用于每个列标题的边框。可以只用一步实现此功能，或者先选择需要设置格式的单元格，这样可以省去许多步骤。为了选择每个单元格，单击 A1 单元，按住 CTRL 键，然后单击 B1、C1 和 D1 单元，每次一个单元格。选择单元格后，把粗边框线型应用于选中单元格的边框。

最后，我们给列标题添加一点颜色。选择 A1:D1，从 Ribbon 选项卡上的字体组里的填充颜色工具的下拉列表中选中一个颜色，如图 2-10 所示。读者可能会注意到，当我们把光标移动某个调色板上的某个颜色时，选择区域的颜色会随之发生变化。给表头选择合适的颜色，使之容易辨认。

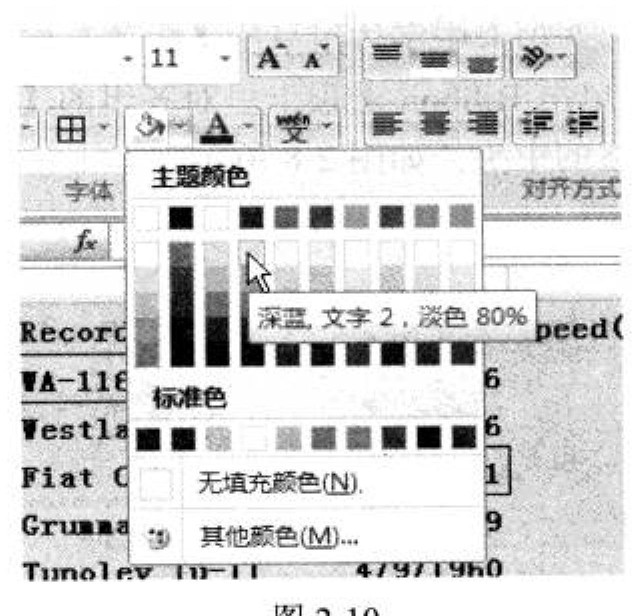


图 2-10

现在这个表格的可读性要比原来的好，而且外观也比较好看，结果如图 2-11 所示。我们在建立电子表格时，即使不打算在报表或演示文稿里使用它们，也要把它们的外观当作一个需要考虑的重要因素。对数据和公式进行分类、组织和并给它们合适的列标头，会大大方便程序的设计、调试、共享，并提高电子表格的可用性。本书自始至终都强调在 Excel 里使用良好的格式。

	A	B	C	D
1	Aircraft Class	Record Setter	Date	Speed (mph)
2	Autogyro	WA-116F	9/18/1986	120.5
3	Rotorcraft	Westland Lynx	8/11/1986	249.1
4	Biplane	Fiat CR42B	1941	323
5	Piston Powered	Grumman F8F Bearcat	8/21/1989	528.33
6	Turboprop Powered	Tupolev Tu-114	4/9/1960	545.07
7	Jet Powered	Lockheed SR-71A	7/28/1976	2193.16
8	Rocket Powered	North American X-15A-2	10/3/1967	4520

图 2-11

Excel 还有一个功能就是数据排序。在我们的表格里，如果把破记录速度按日期排序，我们就会看到比较有意思的结果。为此，选择 A1:D8 单元格，打开 Ribbon 选项卡，单击编辑组，先选择【排序和筛选】命令，然后选择【自定义排序】命令，如图 2-12 所示。

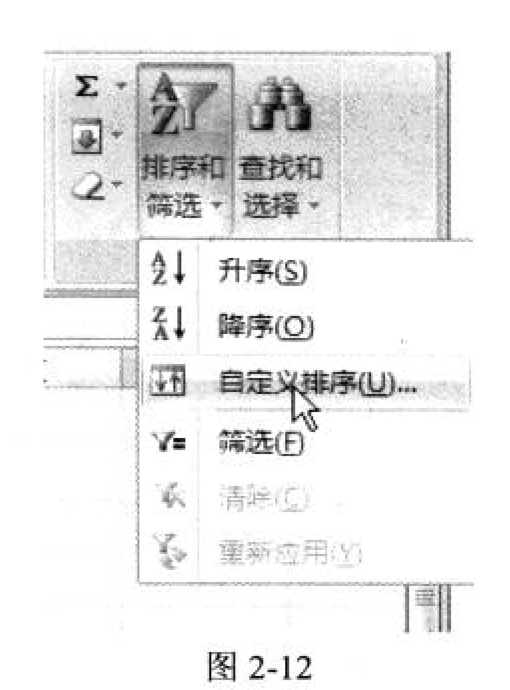


图 2-12

在如图 2-13 所示的【排序】对话框里，从【主要关键字】的下拉列表中选择 Date 选项，再选中右上角的【数据包含标题】这个选项，然后单击【确定】按钮。现在，飞行器速度记录就会按照时间顺序排列。我们很容易看出，双翼飞机保持记录的时间最长，而最近的记录保持者是活塞动力 Grumman F8F Bearcat 飞机。

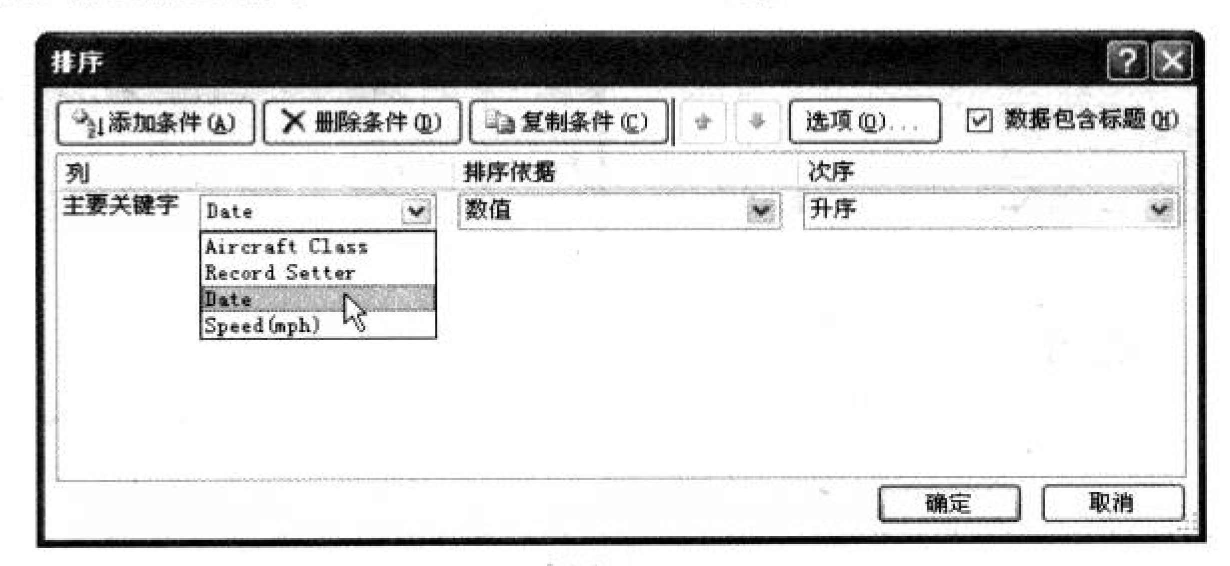


图 2-13

回到前面按时间顺序排列的表格，我们再次使用排序工具，选择按速度排序。还有一种更便捷的办法，Excel 有一个非常有用的撤消和恢复工具。单击快速访问工具栏的撤消工具，如图 2-14 所示，撤消按日期排序的结果。

要养成定期保存编辑结果的良好习惯。要保存这个电子表格，只需要在快捷访问工具栏里单击【保存】图标，如图 2-15 所示。并给它起一个自己喜欢的文件名。

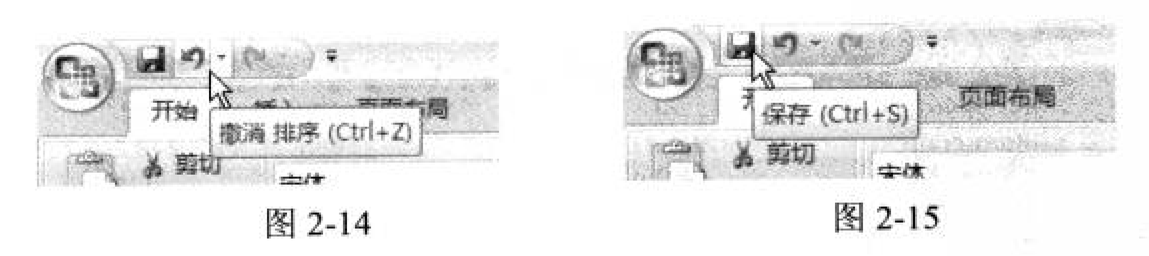


图 2-14　　图 2-15

在 Excel 里设置表格格式还有另一种方法，这种方法对于需要排序和分析的数据表非常有用。现在我们就来介绍这种方法。首先清除表格的格式。选择 A1:D8 单元格，选择清除工具，然后从编辑工具组里选择【清除格式】命令，如图 2-16 所示。



图 2-16

现在确保选中 A1:A8 单元格，从 Ribbon 选项卡里的格式组选择【套用表格格式】，再从中选择其中一个表格格式，如图 2-17 所示。接着，确保选取了【套用表格式】对话框里的【表包含标题】选项，如图 2-18，最后单击【确定】按钮。



图 2-17

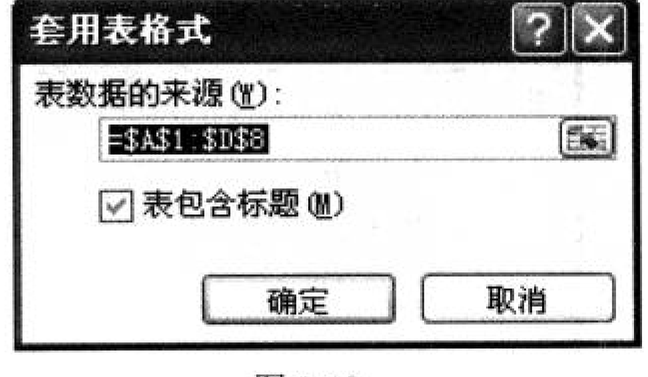


图 2-18

最后得到的结果应该会与图 2-19 相似。但是我们发现日期的格式不对了。当我们最初按“月/日/年”的格式输入日期时，系统会自动认识此日期格式。当表格的格式被清除后，Excel 把格式恢复为普通格式。在普通格式里，日期被显示为一个数值，这个数值就是自 1900 年 1 月 1 日以来的天数。1900 年 1 月 1 日以后，每一天该数值加 1 ，因此，对应于 1900 年 12 月 30 日这个日期，它的值为 366(365 加上闰年多一天)。因此，在表格里显示的数值是自 1899 年 12 月 31 日以后的天数。给日期分配一个数值的好处是，我们可以计算两事件之间的天数，但是需要指出的是，不管显示格式如何，这个值总是存在的。

	A	B	C	D
1	Aircraft Class	Record Setter	Date	Speed (mph)
2	Autogyro	WA-116F	31673	120.5
3	Rotorcraft	Westland Lynx	31635	249.1
4	Biplane	Fiat CR42B	1941	323
5	Piston Powered	Grumman F8F Bearcat	32741	528.33
6	Turboprop Powered	Tupolev Tu-114	22015	545.07
7	Jet Powered	Lockheed SR-71A	27969	2193.16
8	Rocket Powered	North American X-15A-2	24748	4520

图 2-19

现在我们把日期改回“月/日/年”的格式。单击 C 列标号，选择整个 C 列内容，再从 Ribbon 选项卡的数字组里选择【短日期】格式，如图 2-20 所示。



图 2-20

此表格里的日期肯定会被恢复到我们所希望的格式，但是 Biplane 一行例外。这是因为，输入 Biplane 的日期时只有年份。Excel 把这个数值看成是从 1899 年 12 月 31 日以来的天数，因此这个日期变为 1905 年 4 月 24 日。这个问题清楚地表明保持日期格式上的一致性的重要性。由于我们不知道该记录是在 1941 年的哪天创造的，因此有几种选择办法。一种办法是去查找，找出它的准确日期，另一种办法只显示它的年份。首先，我们可以假定它的日期是 1941 年的 6 月 30 日，它是这一年正中间的一个日期。为了输入这个日期，单击错误的 Biplane 记录日期，输入“6/30/1941”，按键盘上的回车键，或者单击公式窗口左侧的输入符号(即勾选符号)。然后，选择包含 Biplane 日期的单元格，从【数学】工具组中的下拉菜中选择【其他的数字格式】，最后，在分类的下拉列表框里选择【自定义】，从类型列表框中选择 yyyy 类型。最后按【确定】按钮，如图 2-21 所示。

新建的数据表在每列的标头单元格都有一个下拉框。单击 Date 列的下拉列表框，选择【升序】排序，如图 2-22 所示。还可以不显示某些日子，只要不选择该日期左侧的选中框就可以了。注意，如果我们没有选中 1986 年，则在表格里不会显示 1986 年创造的两项记录。要显示这两条记录，就要从列标题里的下拉列表中选中 1986 年，或者使用撤消工具。

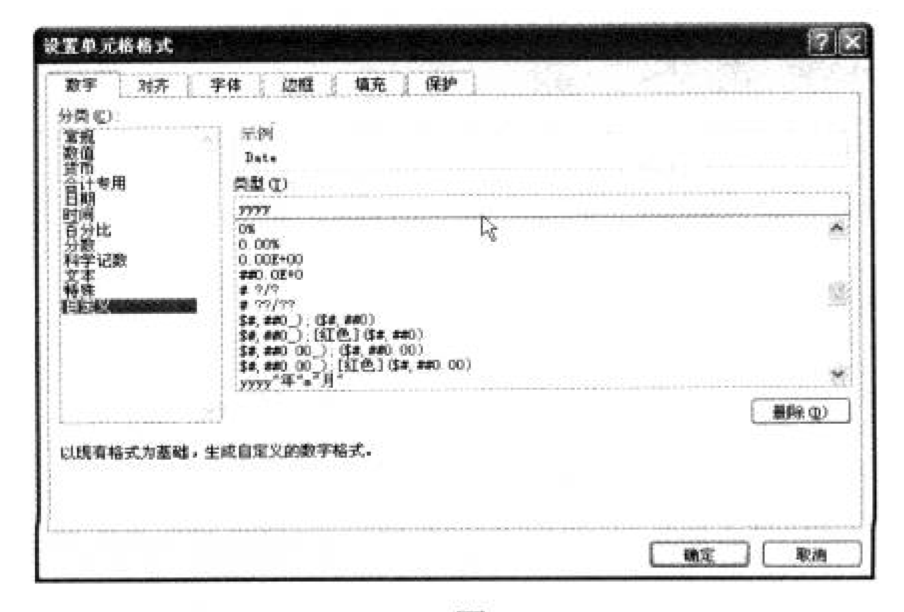






图 2-21　　　　图 2-22

2.3　教程：Excel 的公式输入与设计

在本教程里，我们通过对一个自行车驱动链的分析来学习如何在 Excel 里输入公式，如何执行公式。大多数现代路上比赛用的自行车通过各种组合，可以调节 20 档的速度。参考图 2-23，通常曲柄上有两个链轮，在飞轮上有 10 个链轮。当自行车运动员用踏板驱动曲柄时，链条把链轮的运动传递给飞轮。由于飞轮直接安装在后轮的旋转轴上，因此，飞轮旋转一圈等于后轮旋转一圈。利用一个被称为变速装置的机械设备(图中没有画出)，运动员可以选择链轮与飞轮链轮的各种组合。对于现代道路比赛用的赛车来说，链轮的直径总是大于任意一个飞轮的直径，结果是曲柄旋转一圈会引起后轮旋转多圈。赛车手通常根据实际情形，结合作用于踏板上的作用力和踏板的旋转速度、期望的自行车速度以及道路的起伏变化等因素，选择速度档位(也称 Cadence)。例如，当自行车在平坦的道路上行驶，但准备上坡时，就需要不断增加作用于踏板的力，因为为了克服重力，需要额外的功。假设赛车手想保持均匀的踩踏节奏，就需要选择低档位，这样可以减小施加于踏板上的作用力。然而，我们将发现，低档位也会引起自行车速度的减慢。

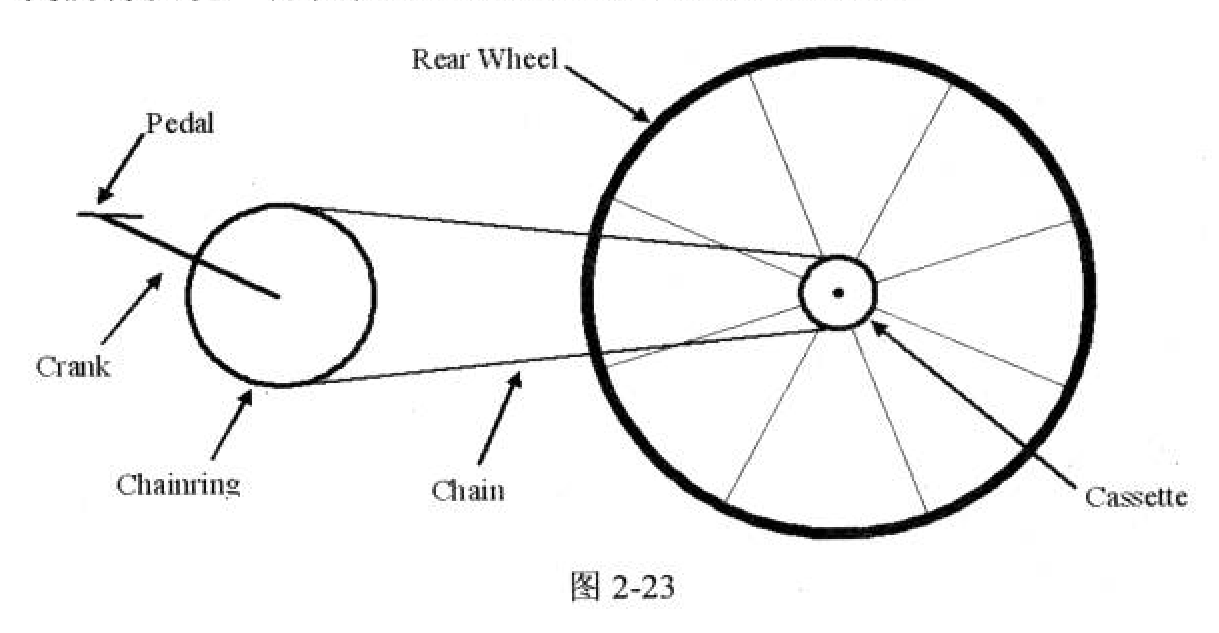


图 2-23

启动 Excel 程序， 如果 Excel 程序已经运行，单击 Office 按钮，再选择新建命令，出现一个如图 2-24 所示的新建工作簿对话框。选择“空工作簿”选项，然后单击“创建”按钮。



图 2-24

通常前轮曲柄与 52 齿或 39 齿的链轮相连。链轮的直径与链轮的齿数成正比。因此有了它们的比率值，我们在计算中就可以只使用齿数而不必考虑链轮的直径。

在 A1 单元里输入 Small Chainring Teeth。

在 A2 单元里输入 Large Chainring Teeth。

在 B1 和 B2 单元里分别输入 39 和 52。

我们用 1~10 个传动数表示传动比，它们代表前链轮与后链轮的直径比，或者给定的链轮的齿数比。传动比表示，脚踏板每旋转一圈，后轮的旋转圈数。在本例里，传动数越小，传动比也越小，自行车脚踏板用力越省。

在 A5 单元里输入 Gear Number，在 A6 和 A7 单元里分别输入 1 和 2。

我们经常需要在 Excel 里建立一个序列。在本例中，需要建立增量为 1 的序列。Excel 能识别这种序列，会自动生成序列。方法是，选择 A6 和 A7，按住选择区域右下角的填充柄如图 2-25 所示，向下拖曳到 A15，就会自动把序列填充到 15。读者要注意，当向下拖动时最后一个单元的值的变化。

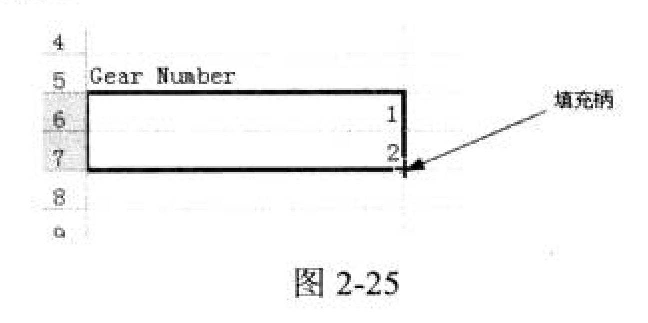


图 2-25

飞轮带有各种大小的齿轮，赛车手根据场地选择飞轮。例如，如果自行车行驶在山地，赛车手希望使用一个大的齿轮，因为大齿轮可以提供爬山所需要的低传动比。本例有 10 种齿轮，它们的齿数分别为 12、13、14、15、16、17、19、21、24 和 27。

在 B5 输入 Cassette Teeth(飞轮的齿数)。

在 B6:B15 输入飞轮的齿数 ，从最大值开始一直到最小值。

现在我们要计算小链轮与飞轮的每个齿轮的传动比。传动比是链轮的齿数与飞轮上的齿数之比。在 Excel 里可以输入一个公式求传动比。

利用 Excel 内置的函数不仅可以执行基本的算术运算，还可以执行高级计算。不管在哪种情形，公式的第一个符号必须是“=”，例如，公式=3＋4×2 表示 4 乘上 2 再加上 3。公式可以包含函数、运算符、引用和常量。Excel 对大小写不敏感。Excel 会自动把公式里的字符变成大写字母。为了保持一致性，本书里所有的 Excel 公式都用大写字母表示。例如，根据半径计算圆的面积的公式可以表示为 PI( )*B4^2。在这个公式里，PI( )是 Excel 求 π 的函数，B4 是单元引用，该单元存放了半径的值。*和^是分别是乘法和指数运算符，2 是常量。Excel 里其他运算符有＋、－和/，分别表示加法、减法和除法运算。还有很多其他数学运算，表 2-1 只是我们在本书中用到的一个常用运算符子集。

表 2-1 数学运算符

运 算 符	符 号	在 Excel 里的用法
加	+	=B2+2.2
减	-	=D2 - C3
乘	*	=5.0*A6
除	/	=C3/D3
指数	^	=D1^2

包括 Excel 在内的大多数编程语言，都用一种标准的层级结构解释数学运算符的运算次序。数学运算符的运算次序的一般规则是：

- 表达式的计算从左到右
- 指数运算的优先级别最高
- 乘法和除法的优先级其次
- 加法和减法的优先级最低

使用括号可以改变运算符的这种标准运算次序。当我们在表达式里使用括号时，运算规则是先计算最里面的括号，从里到外，最后计算最外面的括号。比较好的做法是尽可能多地使用括号，确保表达式按我们的要求进行运算。当用计算工具得到的运算结果与已知结果(来自手工计算或者其他来源)不符时，错误的原因往往与运算次序有关。

在 C5 单元里输入 Small Ring Gear Ration。

在 C6 单元里输入“=B1/B6”公式，并按 Enter 键。这个公式将把 B1 单元的值除以 B6 单元的值。前者表示小链轮的齿数，后者代表飞轮的齿数。除了用键盘输入公式，另一种输入公式的方法是：选择需要输入公式的单元(C6)，并输入“=”，然后单击单元 B1，这相当于选择单元 B1，接着输入“/”，再单击单元 B6，最后按 Enter 键或单击公式窗口的输入按钮(即对勾图标)。通常用这种鼠标单击选择单元的办法比较快，而且不必考虑单元的实际引用地址，因此不容易出错。在本书的后面章节里，需要输入公式时，都采用键盘输入法。但是，我们希望读者在能够熟练操作后，使用鼠标单击方法。

Excel 的一个最重要的功能是它可以很容易执行重复的运算。在下面这个例子里，我们用填充柄把这个公式复制到本列的其他单元格里。

选择 C6，按住填充柄，拖曳到 C15 单元。则 Small Ring Gear Ratio(小链轮传动比)这一列的计算结果如图 2-26 所示。

Cassette Teeth	Small Ring Gear Ratio
27	1.444444444
24	2.166666667
21	0
19	0
17	#VALUE!
16	1.6875
15	1.6
14	1.5
13	1.461538462
12	1.416666667

图 2-26

毫无疑问，读者肯定会发现个别单元格的结果好像不正确。

单击单元 C7，它的公式是“B3/B8”。由于在 B3 单元里没有输入任何值，Excel 就把它当作 0，因此公式运算结果为 0。在这个单元里，真正需要计算的公式是“B1/B8”，当我们把这个公式从一个单元复制到另一个单元时，Excel 会自动根据粘贴单元与复制单元的相对位置关系，修改单元的引用地址。在这个例子里，我们把 C6 里的公式“B1/B6”复制到 C8，因此，单元格的行号增加 2，相应地，公式里的引用单元的行地址也增加 2，因此在 C8 里的公式变为“B3/B8”。在很多情况下，我们希望某个引用地址保持不变。为此需要修改公式，在地址引用的前面添加一个“$”。在复制公式时，如果希望保持行号的引用不变，则要在行号前面加“$”符号。同样道理，如果希望列号的引用保持不变，则在列号的引用之前，加一个“$”符号。在列号和行号引用之前都添加一个“$”符号，则在复制公式时，引用的单元固定不变。

在修改这个错误之前，我们先看看 C10 的内容。在这个单元里出现“#VALUE”内容，表示 Excel 不能计算这个单元里的公式。双击这个单元，我们发现在单元里和公式窗口里都会看到它的公式。我们还发现，Excel 用各种不同的颜色显示被引用单元的边框，且各边框的颜色与公式中引用单元的颜色相同。如图 2-27 所示。

用这种方法显示引用单元为我们查找错误的公式提供了很大的方便。在这里，我们发现这个公式错误引用了单元 B5 的地址，因为 B5 只有一个标题，因此生成了错误的信息。我们可以很容易改正这个错误。**把光标放在 B5 的蓝色边框上，直到出现一个移动箭头为止。如图 2-28 所示，然后把这个边框移动到正确的位置，即 B1 单元。单击公式窗口的【输入】符号，或直接按 Enter 键。**



图 2-27

图 2-28

现在我们来改正全部单元里的公式，固定引用(也叫绝对引用)单元 B1。

双击 C6 单元，然后单击引用单元 B1，如图 2-29 所示。然后按键盘上的 F4 键，Excel 就会在引用单元的列号和行号之前添加“$”符号。注意，每次按 F4 键，单元引用就从单元固定引用、变为行固定引用、再变为列固定引用，再回到相对引用。当我们第一次输入公式时，F4 键也会起作用，因此当我们用鼠标单击输入公式时，F4 键是一个非常方便的工具。

按 Enter 键或单击公式窗口的【输入】按钮，结束公式的输入。选择 C6 单元，然后按住填充柄，重新把正确的公式向下复制到 C15，现在，Small Ring Gear Ratios 这一列的计算结果如图 2-30 所示。

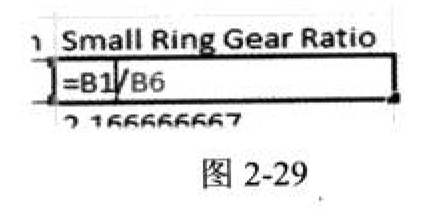


图 2-29

Cassette Teeth	Small Ring Gear Ratio
27	1.444444444
24	1.625
21	1.857142857
19	2.052631579
17	2.294117647
16	2.4375
15	2.6
14	2.785714286
13	3
12	3.25

图 2-30

我们注意到，各单元显示的位数与计算结果有关。Excel 最初用一种默认数值格式显示运算结果。这个常用数值格式最多可以显示 10 位小数(包括小数点在内)。对于非常大的数或非常小的数，如果需要显示 12 位小数，就要把数值显示格式切换到科学表示法。在大多数情况下，我们总需要修改数值显示格式，希望所有单元的计算结果都显示相同的有效位数。改变显示格式并没有改变单元值，因此不会影响后面的计算结果。例如，在本例中，传动比的小数点后第三位开始就没有多大意义，因此，传动比只需要显示两位小数。

选择 C5:C15 单元，然后选择 Ribbon 选项卡上的数字组的【减少小数位数】工具，如图 2-31 所示，直到把小数位数减少到两位为止。

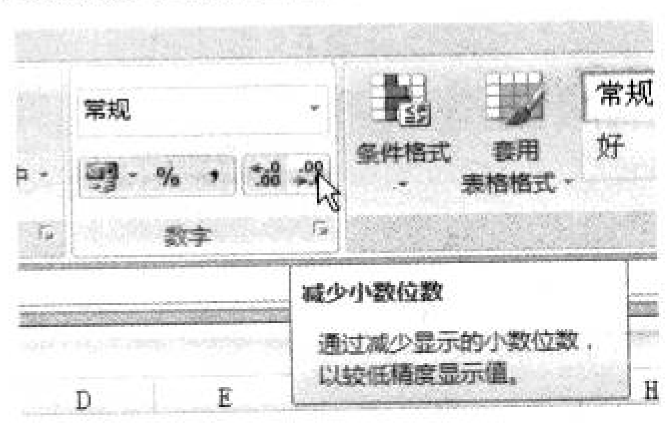


图 2-31

下一步我们要计算大链轮和飞轮的传动比。在 D5 单元里输入 Large Ring Gear Ratio 标题。

在 D6 单元里输入公式“=$B$2/B6”,它表示大链轮与 27 齿飞轮的传动比。接着选择 D6 单元，双击它的填充柄，Excel 会自动把公式在本列里向下复制，复制到邻接列的底部为止。这种双击填充柄的方法比拖动填充柄的方法要快得多，但是只有当邻接列有内容时才会起作用。

选择 D6:D15，用【减少小数位数】的工具，显示小数点后两位数字。

接着，对于每个传动比，我们计算脚踏板每转一圈，自行车骑过的路程。后轮直接与

飞轮耦合在一起。这样，每当飞轮旋转一圈，后轮也旋转一圈。后轮旋转一圈走过的距离等于后轮的周长。或 πD。绝大多数公路自行车的后轮直径为 700mm，因此在这个例子里我们就采用这个数值。由于传动比 GR 代表脚踏板每转一圈，飞轮旋转的圈数。因此自行车旋转一圈走过的距离按方程(2.1)计算：

$$d = GR \times \pi \times D \times \frac{1\text{ft}}{304.8\text{mm}} \tag{2.1}$$

公式中的 1ft/304.8mm 代表转化因子，它把距离从 mm 单位转换为 ft 单位。

在 A3 和 B3 单元里分别输入 Wheel Diameter(mm)和 700。

在 E5 和 F5 单元里分别输入 Small Distance(ft)和 Large Distance(ft)。

要计算 27 齿飞轮的【Small Distance】一列的值，在单元 E6 里输入“=C6*PI( )*$B$3/304.8”公式，用填充柄把这个公式向右复制到 F6，如图 2-32 所示，然后双击填充柄，Excel 同时自动在这两列里把此公式复制到下面的单元里。

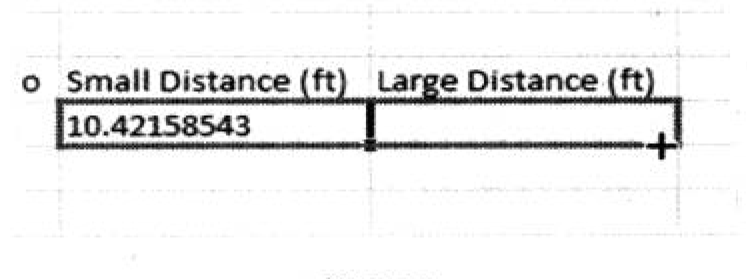


图 2-32

然后，修改数字格式，使得自行车走过的距离显示两位小数。调整全部列的列宽，使得每一列的标题都能正常显示，再把全部数据设置为居中。最后得到的电子表格如图 2-33 所示。

	A	B	C	D	E	F
1	Small Chainring Teeth	39				
2	Large Chainring Teeth	52				
3	Wheel Diameter (mm)	700				
4						
5	Gear Number	Cassette Teeth	Small Ring Gear Ratio	Large Ring Gear Ratio	Small Distance (ft)	Large Distance (ft)
6	1	27	1.44	1.93	10.42	13.90
7	2	24	1.63	2.17	11.72	15.63
8	3	21	1.86	2.48	13.40	17.87
9	4	19	2.05	2.74	14.81	19.75
10	5	17	2.29	3.06	16.55	22.07
11	6	16	2.44	3.25	17.59	23.45
12	7	15	2.60	3.47	18.76	25.01
13	8	14	2.79	3.71	20.10	26.80
14	9	13	3.00	4.00	21.64	28.86
15	10	12	3.25	4.33	23.45	31.26
16						

图 2-33

本例子最后需要计算自行车的速度。显然，自行车的速度与所选的档位有关，与脚踏板的踩踏速度有关。假设脚踏板踩踏速度，即踏频用 N 表示，通常用每分钟多少圈(RPM)

表示，为了保持踏频的稳定，赛车手根据地形、风速及其他条件调整档位。常见的踏频是90RPM，如果自行车的速度用每小时英里数表示，则它可由方程(2.2)求得：

$$速度 = N \times d \times \left(\frac{60\,\text{min}}{1\text{hr}}\right) \times \left(\frac{1\text{mile}}{5280\text{ft}}\right) \tag{2.2}$$

公式中 60min/hr 和 1mile/5280ft 都是转换因子，这两个因子是把速度的单位从 ft/min 转换为 mph 所必需的。

现在在第 4 行里插入踏频作为输入变量。然而一个良好的习惯是在电子表格里功能不同的两个数据区之间有空行或空列。在本例，左上角的一个小表格存放了链轮的齿数、后轮直径等输入数据。底部的大表格保存了计算结果。当然实际上，我们也可以在 E 列和 F 列的公式里直接输入后轮的直径 700 这个数值，但是如果真的这样做，当需要分析后轮直径为 650mm(这是三项全能常用的自行车)的情况，则要修改 E6:F15 单元，把所有的 700 改为 650，这实在是太麻烦了。如果在单元 B3 为后轮的直径定义一个变量，则只要在这个单元里修改后轮的直径，就可以计算使用其他直径后轮的自行车的速度。

根据需要我们可以插入行或列，更重要的是，Excel 会自动修改公式中的引用地址，这样能保证公式的正确性。**要另外添加第 4 行，首先选择第 4 行，方法是用鼠标单击电子表格左侧的第 4 行的行号，如图 2-34 所示。**

接着，选择单元组里的【插入单元格】工具，如图 2-35 所示，Excel 就插入一行。

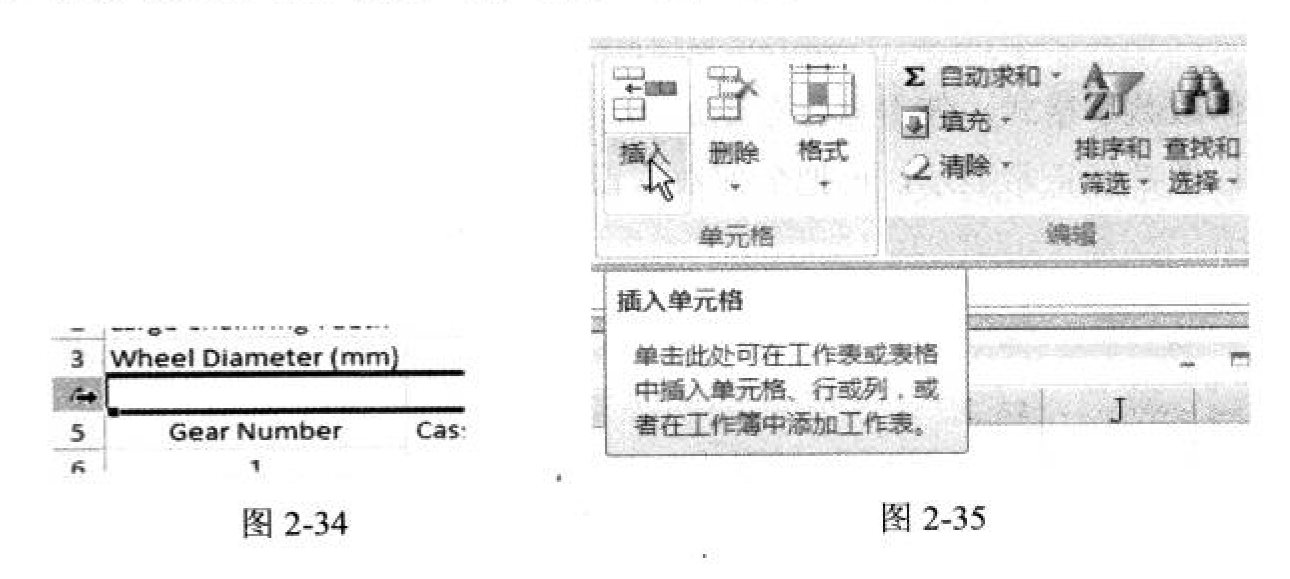


图 2-34　　　　图 2-35

在 A4 和 B4 单元里分别输入 Cadence(RPM)和 90。

在 G6 和 H6 单元里分别输入 Small Speed(mph)和 Large Speed (mph)，然后在 G7 单元里输入公式“=$B$4*E7*60/5280”，这个公式将计算 27 齿齿轮的小链轮的速度。接着，用填充柄把这个公式向右复制，并向下复制。然后修改数值格式，只显示两位小数，并调整列宽，使得列标题能正确显示，把全部数据设置为居中。最后，给这两个表格添加边框，并给表格标题添加背景颜色，并且保存电子表格。最后生成的电子表格类似于图 2-36。

	A	B	C	D	E	F	G	H
1	Small Chainring Teeth	39						
2	Large Chainring Teeth	52						
3	Wheel Diameter (mm)	700						
4	Cadence (RPM)	90						
5								
6	Gear Number	Cassette Teeth	Small Ring Gear Ratio	Large Ring Gear Ratio	Small Distance (ft)	Large Distance (ft)	Small Speed (mph)	Large Speed (mph)
7	1	27	1.44	1.93	10.42	13.90	10.7	14.2
8	2	24	1.63	2.17	11.72	15.63	12.0	16.0
9	3	21	1.86	2.48	13.40	17.87	13.7	18.3
10	4	19	2.05	2.74	14.81	19.75	15.1	20.2
11	5	17	2.29	3.06	16.55	22.07	16.9	22.6
12	6	16	2.44	3.25	17.59	23.45	18.0	24.0
13	7	15	2.60	3.47	18.76	25.01	19.2	25.6
14	8	14	2.79	3.71	20.10	26.80	20.6	27.4
15	9	13	3.00	4.00	21.64	28.86	22.1	29.5
16	10	12	3.25	4.33	23.45	31.26	24.0	32.0

图 2-36

仔细分析运算结果，我们发现，20 速的道路自行车并没有 20 个不同的传动比。仔细分析本例的传动比，对于小链轮，传动比从 1.44 变化到 3.25。对于大链轮，传动比从 1.93 变化到 4.33。虽然，只有一个传动比(3.25)是两种链轮共有的，但是大多数赛车手使用的传动比不会超过 14 种。自行车车手在启动时使用最低传动比。当准备加速时，就通过较小飞轮的切换，选择较高的传动比。通常在小链轮的第 8 档而不是第 9 档时，赛车手切换到大的链轮，接着马上把飞轮的档位向下调整 3 档，定位在大链轮的第 5 档上(实际上是把传动比从 2.79 切换到 3.06)。在这种情况下，赛车手利用了小链轮的前 8 档和大链轮的后 6 档。实际上他一共只使用了 14 档。

2.4　教程：Excel 内置函数的使用

Excel 里内置了很多函数，这些函数用来解决工程问题非常有用。如像 e^x 和 ln(*x*) 等标准数学函数，sin( )和 cos( )等三角函数，求平均值和标准差等统计函数和 IF-ELSE 等逻辑判断函数。

每个函数的调用都有一定的格式或语法。所有函数的最前面必须有一个“=”，除非一个函数嵌入另一个函数内部或表达式里。一般函数的调用要根据以下的格式或语法：

```
FUN_NAME(arg1,arg2,…)
```

这里，FUN_NAME 是函数名，arg1,arg2，…是函数的参数。一个函数通常有一个或多个参数，但是在前面的例子里，PI( )函数没有参数。在后面的练习中，我们将学习有关函数格式的详细内容。

例 2.1

用函数计算$y=e^x$在x=0,1,2,3 的值。

解：

分别在 A1 和 B1 输入 *x* 和 *y*。

在 A2:A5 单元里分别输入 0,1, 2,3。

选择 B2，然后单击 Ribbon 的公式选项卡，再从函数库中选择数学与三角函数，然后

向下找到 EXP( )函数，如图 2-37 所示。

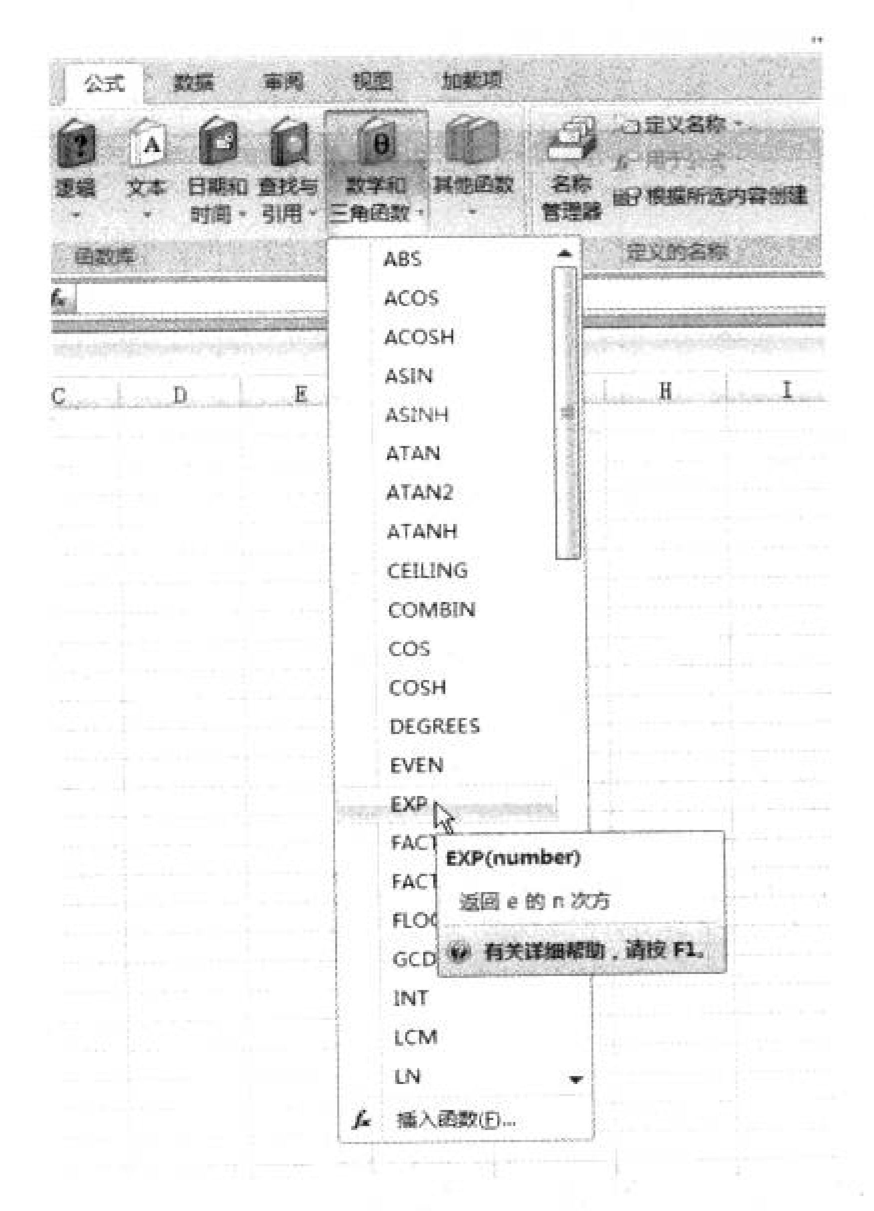


图 2-37

注意，我们可以用右侧的滑动条浏览这一组的全部函数。当把光标悬浮在某个函数上面时，Excel 就会出现一个窗口显示该函数的用法及作用。要想知道某个函数的详细信息，只要按键盘上的 F1 键，Excel 就会显示该函数的详细说明。

在 EXP( )函数参数对话框里，我们要为 Number 参数输入一个数或单元的引用。在这个例子里，我们要用到单元的引用，输入 A2，然后按【确定】按钮，如图 2-38 所示。

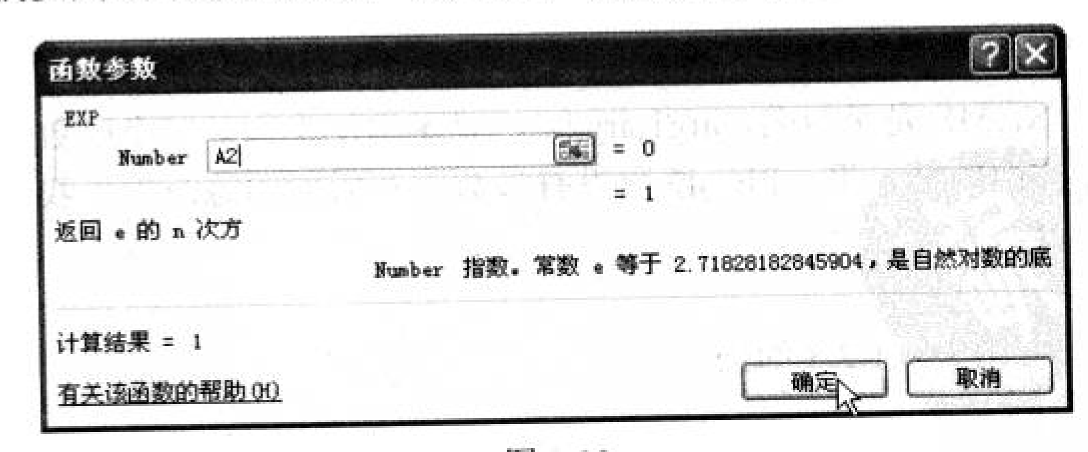


图 2-38

利用填充柄把这个公式向下复制到单元 B5。把列标题设置为居中，把 A2:A5 单元的 x 值设置为居中。给列标题添加双下划线。最后得到的电子表格如图 2-39 所示。

	A	B
1	x	y
2	0	1
3	1	2.718282
4	2	7.389056
5	3	20.08554

图 2-39

例 2.2

计算 0º~360° 范围内每隔 5º 的所有角度的正弦、余弦和正切值。

解：

在 A1 单元里输入 Angle,degrees(角度，度)。

在 B1:D1 单元里分别输入 Sine，Cosine，Tangent 三个列标题。

下一步我们打算使用一种稍微不同于填充柄的方法，输入 0º~360° 范围每隔 5° 的角度序列。这种方法更适合于输入大型序列。

在 A2 单元里输入 0。选择 A2，从 Ribbon 里选择开始选项卡，从编辑组中选择【填充】工具，接着选择【系列】命令，如图 2-40 所示。

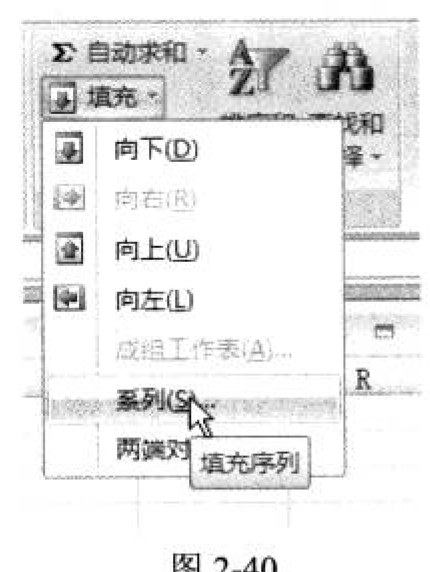


图 2-40

接下来会弹出一个【序列】对话框。把【序列产生在】设置为【列】。把【类型】属性设置为【等差序列】，把【步长值】设置为 5，把【终止值】设置为 360。然后单击【确定】按钮，如图 2-41 所示。

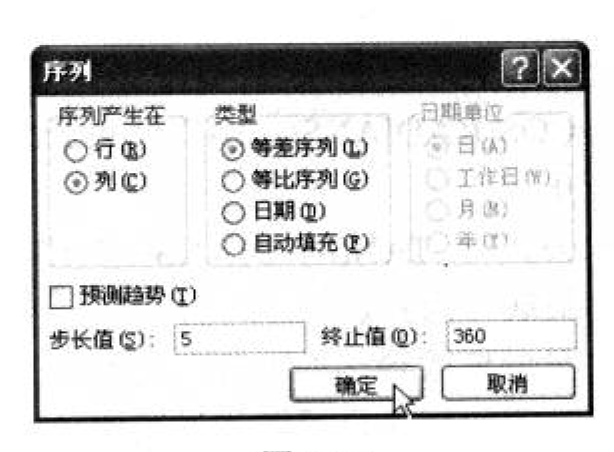


图 2-41

现在我们输入公式求正弦的值。求正弦函数的语法是：

```
SIN(number)
```

这里的 *number* 是角度的弧度。由于这里的角度是用度表示的，因此需要把它转换为弧度。180° 的弧度值是 π 。

我们也可以不用 Ribbon 的公式选项卡输入函数。对于常用的函数，在单元里用键盘输入函数通常会更加方便些。本教程就要介绍这种方法。

在单元 B2 里用键盘输入“=SIN(A2*PI( )/180)”公式。按 Enter 键结束公式的输入，然后拖动填充柄填充到表格的最底行。

注意，当我们输入“=s”时，会出现一个下拉对话框，显示所有以 s 开头的函数，如图 2-42 所示。当我们无法记住函数的准确名字时，这个下拉列表框就会非常有用。双击弹出对话框里的函数时，该函数就会自动输入到单元中。

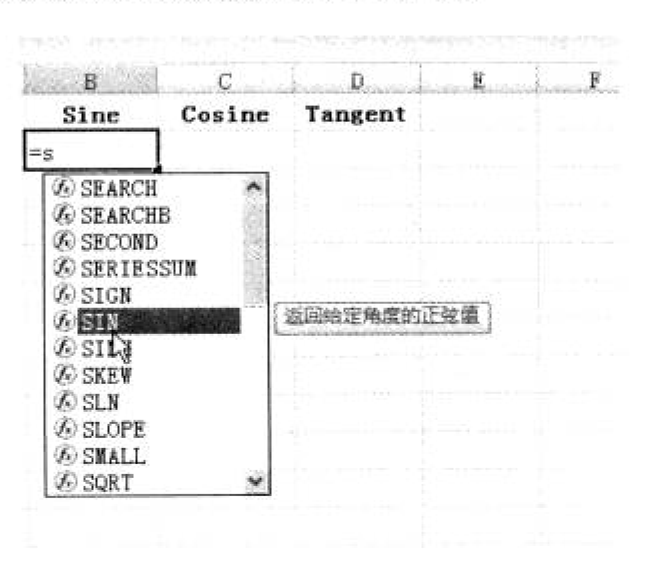


图 2-42

另外我们还需要注意到，一旦函数输入到单元里，在输入右括号之前，会弹出一个提示框，说明函数的语法，如图 2-43 所示。如果函数需要多个参数，这个提示就非常有用。

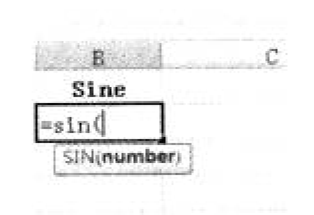


图 2-43

在 Excel 里有一个函数可以把度转换为弧度。在求余弦值，我们不打算使用前面的 PI( )/180 这种转换方法而是使用这个函数。

在 C2 单元里输入公式“=COS(RADIANS(A2))”,结束公式的输入后用填充柄把公式向下复制，直到表格的最下一行止。

接着，在 D2 单元里输入公式“=TAN(RADIANS(A2))”，结束公式的输入，用填充柄把公式向下复制，直到表格的最底行为止。

在结束此电子表格之前，我们要改善它的外观。把整个表格设置为居中。调整列宽，使得表格的列标题能正常显示。把数值的显示格式设置为 4 位小数。

读者肯定会注意，90º 和 270º 的正切是一些很大的数值。三角函数的计算原理表明，当角度趋近 90º 或 270º 时，它们的正切值趋近无穷大。在 90º 或 270º 时，我们可以认为它

们的正切函数没有定义。为此，要删除在相对应的两个单元里的函数。**右击 D21 单元，然后从快捷菜单中选择【清除内容】命令，如图 2-44 所示。对 D57 单元，重复同样的操作。然后再次调整 D 列的宽度。**

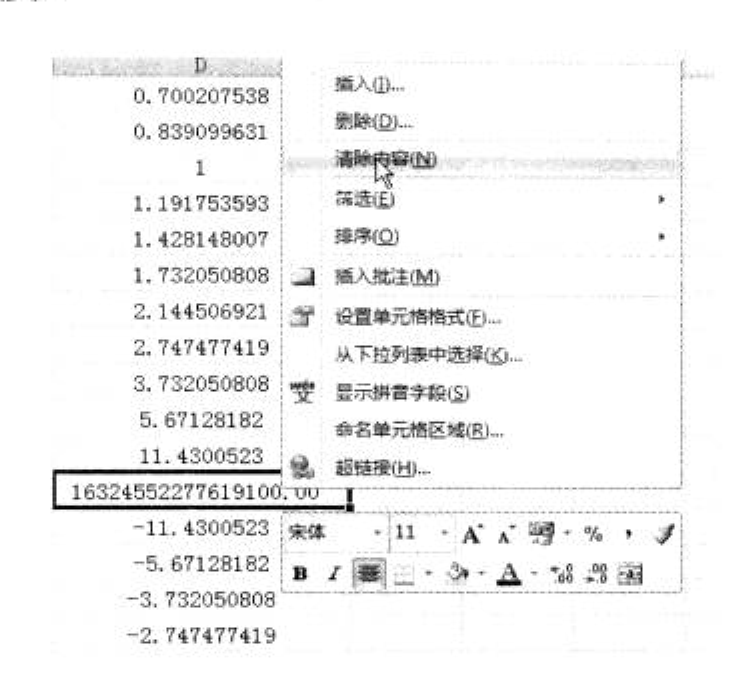


图 2-44

下面我们给这个表格插入一个标题。为了插入一行，首先选择第一行，通过鼠标右击选择【插入】命令，如图 2-45 所示。



图 2-45

选择 A1:D1 单元，然后从 Ribbon 工具栏上的对齐组选择【合并后居中】工具，如图 2-46 所示。

现在在 A1 单元里输入 Trigonometric Functions。之后我们给此标题、列标题和整个表格添加一个粗线框，再给列标题设置一种填充颜色。最后得到的电子表格如图 2-47 所示。



图 2-46

Trigonometric Functions			
Angle, degrees	Sine	Cosine	Tangent
0	0.0000	1.0000	0.0000
5	0.0872	0.9962	0.0875
10	0.1736	0.9848	0.1763
15	0.2588	0.9659	0.2679
20	0.3420	0.9397	0.3640
25	0.4226	0.9063	0.4663
30	0.5000	0.8660	0.5774
35	0.5736	0.8192	0.7002
40	0.6428	0.7660	0.8391
45	0.7071	0.7071	1.0000
50	0.7660	0.6428	1.1918
55	0.8192	0.5736	1.4281
60	0.8660	0.5000	1.7321
65	0.9063	0.4226	2.1445
70	0.9397	0.3420	2.7475
75	0.9659	0.2588	3.7321
80	0.9848	0.1736	5.6713
85	0.9962	0.0872	11.4301
90	1.0000	0.0000	
95	0.9962	-0.0872	-11.4301
100	0.9848	-0.1736	-5.6713

图 2-47

例 2.3

在滚球轴承的制造过程中，各部件都需要经过一道硬化工序。具体过程是先加热再迅速冷却，或者把它浸入到油槽或水槽里，这个过程即为淬火。滚球轴承的温度是时间的函数。$T(t)$可以按方程(2.3)估算：

$$T(t) = (T_i - T_\infty)e^{-t/\tau} + T_\infty \tag{2.3}$$

式中t是轴承浸入槽中的时间，单位为秒。T_i是轴承的初始温度，T_∞是油的温度。τ是时间常数，单位是s，它与轴承的材料、轴承的几何形状及油的特性有关。要求建立一个电子表格，以T_i、T_∞和τ为输入变量，计算在 0~10s时间里轴承的温度随时间的变化。假设时间常数τ=40s。

解：

在 A1:B3 的单元里分别输入“Ti，degrees C”，“Tinf，degrees C”，“Tau，seconds”和“600”，“30”，“40”。并设置它们的格式，如图 2-48 所示。

	A	B
1	Ti, degrees C	600
2	Tinf, degrees C	30
3	Tau, seconds	40
4		

图 2-48

在 D1 和 E1 单元里分别输入“time，seconds”和“Temp，degrees C”，在 D2:D12 单元里用填充方法输入 0~10 共 11 个整数。并把合适的格式应用于此表格上，如图 2-49 所示。

在单元 E2l 输入公式“=($B$1-$B$2)*EXP(-D2/$B$3)+$B$2”，并利用填充柄把它向下复制到单元 E12。把表格设置为合适的格式，如图 2-50 所示。

D	E
time, seconds	Temp, degrees C
0	
1	
2	

图 2-49

D	E
time, seconds	Temp, degrees C
0	600.0
1	585.9
2	572.2
3	558.8
4	545.8
5	533.0
6	520.6
7	508.5
8	496.7
9	485.2
10	473.9

图 2-50

经过对淬火过程深入分析，正如问题所描述的那样，轴承的初始温度是 600°C。许多工程问题的初始条件和最终状态的结果都是已知的。这些已知条件对于判断计算机求解的正确性非常有用。在这个问题里，我们知道，经过长时间的淬火后，轴承的温度接近于油的温度。只要在 D12 单元里输入一个很大时间，就可以验证这个结果。

在 D12 单元里输入“300”，即可以计算经过 300s 的冷却后，轴承的温度将会是多少。答案是 30.3°C。

我们这个问题要求把轴承冷却到 60°C 以下所需要的时间。确定这个时间的方法有若干种。在本例里，我们采用缩小时间范围、放大、再缩小范围再放大的搜索方法找到准确的解。其他更好的求解方法，如图形法，将在本书的第 5 章和第 6 章的目标搜索等专用工具中介绍。

在本例里，我们已经确定答案是在 0~300s之间。为了更好地确定时间范围，我们把时间步长设置为 30s。**在D2 和D3 单元里分别输入“0”和“30”，并用填充柄向下复制到D12。**

现在我们发现，在 90s~120s 之间，轴承的温度降低到 60°C 以下。现在把时间的步长设置为 3s，放大这个时间范围。**在 D2 和 D3 单元里分别输入“90”和“93”,并用填充柄向下复制到 D12。**

现在我们可以看出，大概在 117s 之后，轴承的温度降低到 60°C 以下。如果要求更加准确的结果，则可以再次放大这个时间范围，把时间步长设置为 0.3s。**更准确的答案是 117.9°C。**

2.5 教程：用 IF 执行逻辑测试

在许多工程问题的分析中，我们需要判断一个数或一个字符串的值，根据判断结果，选择不同的路径执行运算。Excel 内置了几个逻辑判断函数，这些函数可以进行比较，并根据比较结果返回一个值。其中一个最有用的逻辑判断函数是 IF 语句。IF 语句的用法是：

```
IF(logical_test, value_if_true, value_if_false)
```

其中 logical_test 是条件测试，它可以是任何值或任何运算结果为 true 或 false 的表达式，这些表达式由=(等于)、<(小于)、>(大于)、≤(小于等于)、≥(大于等于)和<>(不等于)这 6 个比较运算符组成。当 logical_test 条件测试结果为真(true)时，IF 函数返回参数 value_if_true 值；当 logical_test 条件测试结果为假(true)时，IF 函数返回参数 value_if_false 值。返回的值可以是数值、字符串或另一个公式的值。

在本教程里，我们将学习如何在以下几个例子里使用 IF 函数。

例 2.4

用 IF 函数计算一个数的绝对值。

解：

一个数的绝对值是这样规定的：如果这个数大于或等于 0，则它的绝对值就是它本身，如果这个数小于 0，则它的绝对值是这个数乘以 – 1。

在 A1 和 A2 单元格里分别输入 Number 和 Absolute Value。调整列宽到最合适位置。

在 B1 单元里输入“5”。

在 B2 单元里输入公式“=IF(B1<0，–B1，B1)”，单元 B2 的结果是 5。

现在测试这个 IF 语句是否正确。在 B1 单元里输入“ – 3”，B2 单元的结果应该是“3”。

例 2.5

在机械制造中，对相互配对的零件尺寸的误差要求非常严格。通常的做法是，使用坚硬的钢榫钉，准确定位到模型的另一半上，这个模型可以用注射成型，如图 2-51 所示。每次浇注一个零件后，必须把它的两部分拆开来。在最初用车床加工这个模具时，定位孔的直径必须非常准确。如果直径太小，模型很难装配起来，也很难拆卸下来。如果直径太大了，定位不准，制造出来的零件与模型不匹配。在本练习里，我们将用 IF 语句接受或拒绝定位孔的直径测量值。

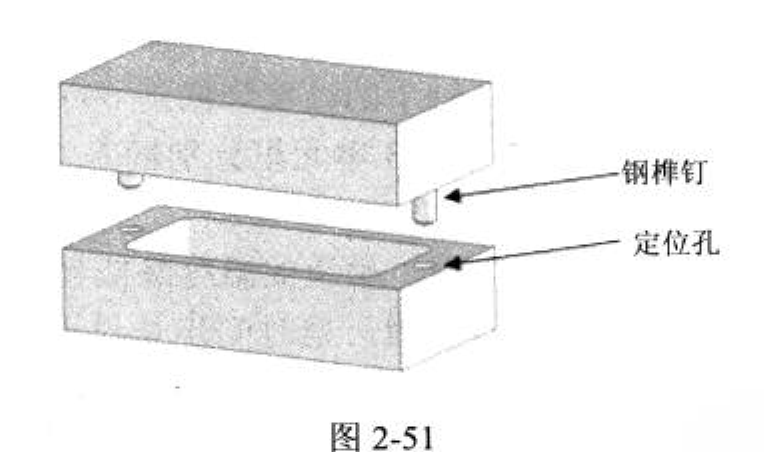


图 2-51

定位孔直径的标准是 0.2500＋0.0010、– 0.0005。它的意思是，直径在 0.2495in~0.2510in 之间的定位孔都是合格的。利用 IF 语句实现：

(1) 如果定位的直径符合标准，接受。

(2) 如果定位孔直径太大，因为太大而拒绝(Reject High)。

(3) 如果定位孔直径太小，因为太小而拒绝(Reject Lower)。

解：

我们尽量把电子表格做成通用表格，这样，当需要修改定位孔的目标直径或公差时，

就不必重写全部的公式。首先用变量表示这些值。

在 A1:A3 单元里分别输入 Target Diameter(in)，Plus Tolerance(in)，Minus Tolerance(in)，然后在 B1:B3 里分别输入 0.25，0.0010，－0.0005。

在 A5:B5 单元里分别输入 Measured Diameter (in)和 Tolerance Check。

在 A6:A25 输入以下测量得到的直径：0.2501，0.2496，0.2498，0.2512，0.2502，0.2512，0.2495，0.2499，0.2500，0.2493，0.2496，0.2506，0.2502，0.2499，0.2506，0.2503，0.2496，0.2494，0.2497，0.2508。

用 IF 语句求解本例有好几种方法。我们使用下面这种方法：首先，我们把测量的直径值与标准直径进行比较，得到它们的差值。然后用 IF 语句判断这个差值是否大于正公差，如果大于正公差，IF 语句返回 REJECT HIGH。如果不满足，我们用第二个 IF 语句判断此差值是否小于负公差。如果小于负公差，则返回 REJECT LOW，最后，两个条件都不满足，则表示定位孔的直径符合公差的要求，返回 ACCEPT。

在 B6 单元里输入"=IF((A6-$B$1)>$B$2,"REJECT HIGH",IF((A6-$B$1)<$B$3,"REJECT LOW","ACCEPT"))"公式。公式中的引号表示一个字符串，而且是必不可少的。用填充柄向下复制到 B7:B25 单元里。

调整数字格式，全部数据都显示 4 位小数，并把列宽调整到合适宽度，使得列标量都能正常显示。把全部数据设置为居中。最后得到的电子表格如图 2-52 所示。

	A	B
1	Target Diameter (in)	0.2500
2	High Tolerance (in)	0.0010
3	Low Tolerance(in)	-0.0005
4		
5	Measured Diameter (in)	Tolerance Check
6	0.2501	ACCEPT
7	0.2496	ACCEPT
8	0.2498	ACCEPT
9	0.2512	REJECT HIGH
10	0.2502	ACCEPT
11	0.2512	REJECT HIGH
12	0.2495	ACCEPT
13	0.2499	ACCEPT
14	0.2500	ACCEPT
15	0.2493	REJECT LOW
16	0.2496	ACCEPT
17	0.2506	ACCEPT
18	0.2502	ACCEPT
19	0.2499	ACCEPT
20	0.2506	ACCEPT
21	0.2503	ACCEPT
22	0.2496	ACCEPT
23	0.2494	REJECT LOW
24	0.2497	ACCEPT
25	0.2508	ACCEPT

图 2-52

现在我们得到的 Excel 函数统计定位孔的判断结果为 ACCEPTED，REJECT HIGH 和 REJECT LOW 的次数。很显然，对于较小的数据集，用手工统计，也不是一件难事。但是想象一下，如果有数千个测量数据需要统计，用手工方方法需要多长时间。COUNTIF 函

数的用法如下：

```
COUNTIF(range,criteria)
(COUNTIF(范围,条件))
```

函数中的 range 表示一组含有需要统计数据的单元格，criteria 是以数值、表达式或字符串的形式定义统计单元。

建立一个小表格，输入三个统计标准：ACCEPTED、REJECT HIGH 和 REJECT LOW 的个数和相应的比例。

在 D2:D4 单元格分别输入 Accepted，Rejected High，Rejected Low 三个标题。

在 E1 和 F1 单元里分别输入 Number 和 Percent 标题。

在 E2 单元格里输入“=COUNTIF(B6:B25，"ACCEPT")” 公式。公式中的引号表示一个字符串，是必不可少的。统计被判断为 REJECT HIGH 和 REJECT LOW 的零件个数，分别在 E3 和 E4 输入“=COUNTIF(B6:B25,"REJECT HIGH")”公式和“=COUNTIF(B6:B25,"REJECT LOW")”公式。

在 F2 单元里输入公式“=E2/SUM($E$2:$E$4)”，并用填充柄把它复制到 F3:F4 单元里。要把计算结果显示为百分比的形式，选择 F2:F4 单元，单击 Ribbon 上的数字组里的【百分比样式】，如图 2-53 所示。

调整列宽，并把数据居中，最后得到的结果类似于图 2-54。

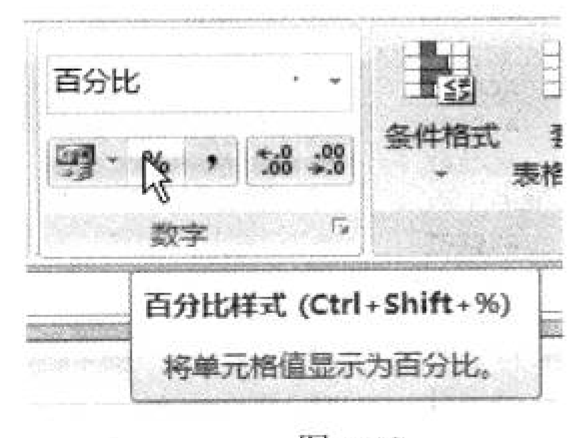


图 2-53

D	E	F
	Number	Percent
Accepted	16	80%
Rejected High	2	10%
Rejected Low	2	10%

图 2-54

Excel 可以根据单元的值自动改变单元的显示格式。我们可以利用 Excel 的这个功能，改善表格的显示效果。在这个例子里，我们根据 B6:B25 单元里的文本内容，把那些被拒绝的零件突出显示出来，这要用到条件格式。

选择 B6:B25，然后选择 Ribbon 的样式组里的【条件格式】命令。选择【突出显示单元格规则】|【文本包含】命令，如图 2-55 所示。

接着，出现一个【文本中包含】对话框。在此对话框的格式单元里输入 ACCEPT，然后选择【绿填充色深绿色文本】选项作为单元的格式。如图 2-56 所示，单击【确定】按钮，关闭此对话框。

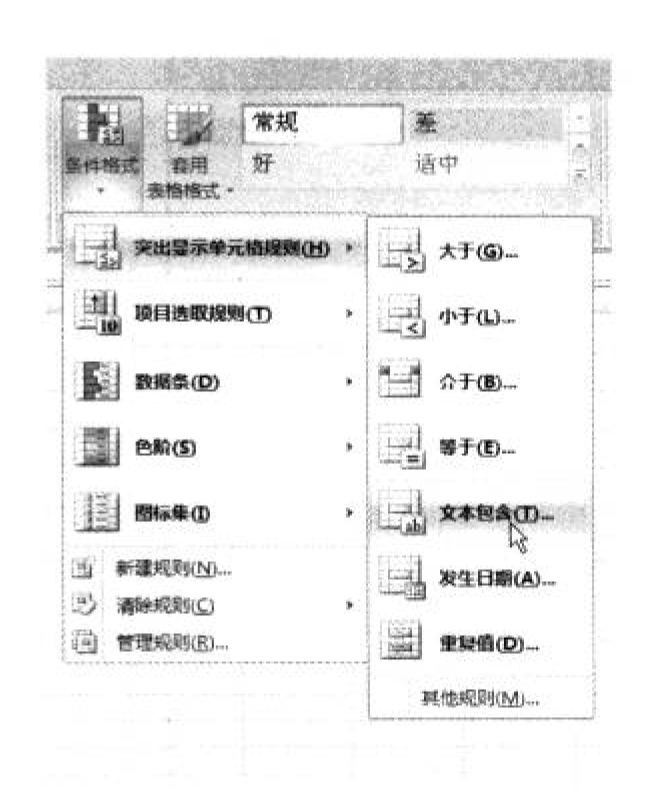


图 2-55

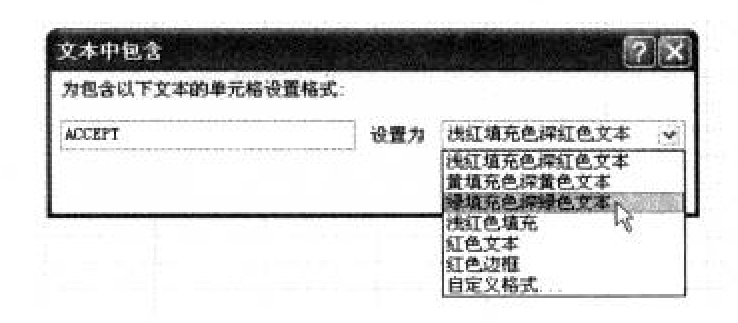


图 2-56

接着我们再次重复条件格式的步骤，应用第二个条件格式。在包含文本对话框的格式单元的编辑框里输入 High，再选择【浅红填充色深红色文本】的格式作为单元格式。

最后，重复上述的条件格式的步骤，应用第三个条件格式。在【文本中包含】对话框的格式单元里输入 Low，选择【自定义格式...】，出现一个【设置单元格格式】对话框，选择【字体】选项卡，再从颜色下拉列表中选择深蓝色，如图 2-57 所示。

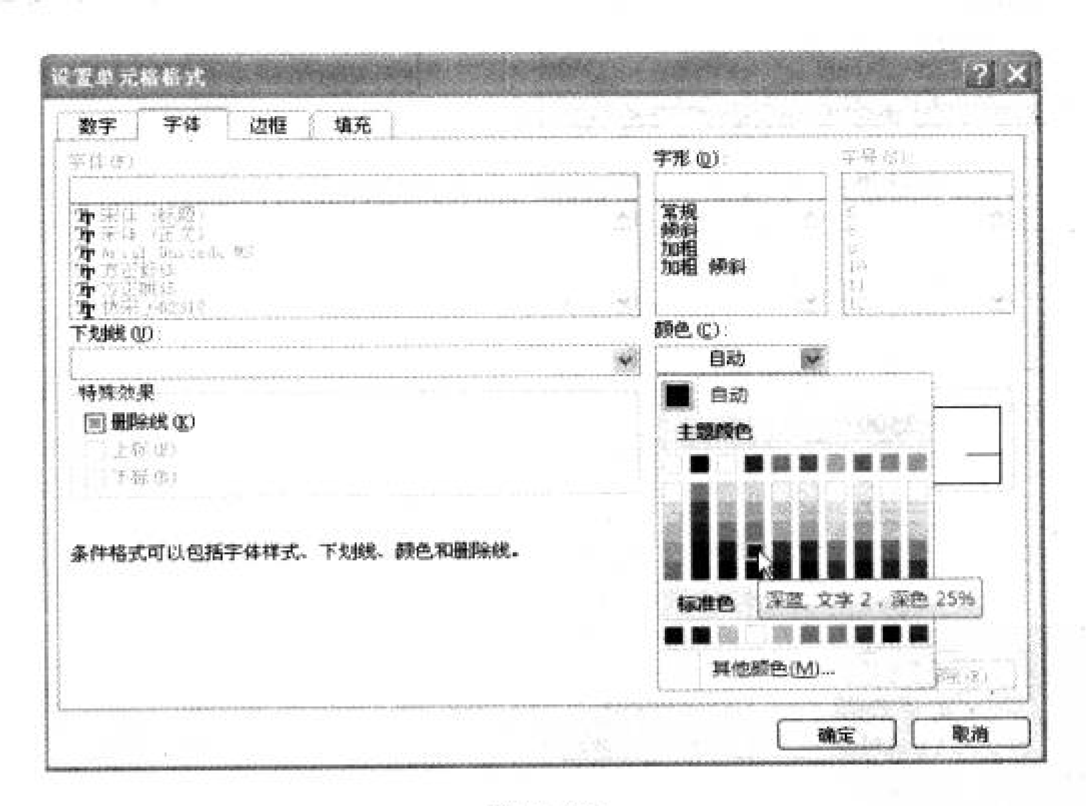


图 2-57

接着，选择【填充】选项卡，再选择淡蓝色为单元格的背景颜色，如图 2-58 所示。单击【确定】按钮，关闭此对话框。

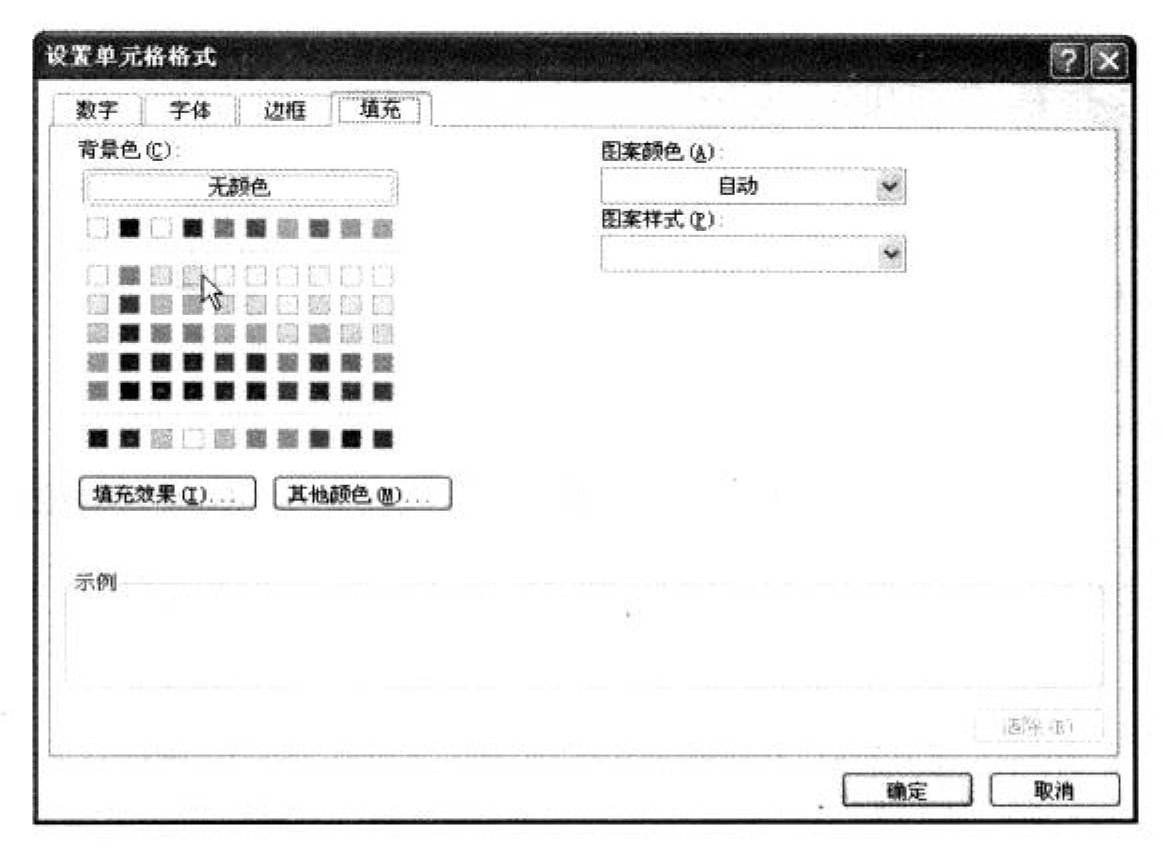


图 2-58

在结束此电子表格之前，为每个表格添加边框。突出显示表格的标题，对齐表格中的数据，使表格的外表美观，最后得到的电子表格如图 2-59 所示。

	A	B	C	D	E	F
1	Target Diameter (in)	0.2500			Number	Percent
2	High Tolerance (in)	0.0010		Accepted	16	80%
3	Low Tolerance(in)	-0.0005		Rejected High	2	10%
4				Rejected Low	2	10%
5	Measured Diameter (in)	Tolerance Check				
6	0.2501	ACCEPT				
7	0.2496	ACCEPT				
8	0.2498	ACCEPT				
9	0.2512	REJECT HIGH				
10	0.2502	ACCEPT				
11	0.2512	REJECT HIGH				
12	0.2495	ACCEPT				
13	0.2499	ACCEPT				
14	0.2500	ACCEPT				
15	0.2493	REJECT LOW				
16	0.2496	ACCEPT				
17	0.2506	ACCEPT				
18	0.2502	ACCEPT				
19	0.2499	ACCEPT				
20	0.2506	ACCEPT				
21	0.2503	ACCEPT				
22	0.2496	ACCEPT				
23	0.2494	REJECT LOW				
24	0.2497	ACCEPT				
25	0.2508	ACCEPT				

图 2-59

现在我们再简单地说明一下给变量赋值的重要作用。假设我们有很大的零件库，保存以前的测量结果。现在有一个客户要求订购一批特殊的零件，要求零件的公差在+0.001 00 和 - 0.000 0 之间。那么，这个零件库有多少零件能够立即发货且满足这个客户要求？要回

答这个问题，只需要在 B3 单元里输入 0.000 0，结果是 40%。

例 2.6

一个检测程序需要连续三次测试部件。每个零件都分配一个序列号，每次测试结果是 Pass 或 Fail。现在假设接受零件的标准有两条：(1)只有三次测试都合格的部件才可以被接受；(2)至少有一次通过测试的零件才可以接受。分别根据这两个准则，用 Excel 判断表 2-2 中哪些零件是符合标准的，哪些零件不符合标准。

表 2-2

零件序号	测试 1	测试 2	测试 3
1	Pass	Fail	Fail
2	Pass	Fail	Pass
3	Pass	Pass	Pass
4	Fail	Fail	Fail
5	Fail	Pass	Pass

解：

在这个例子里，我们要用嵌套 IF 语句进行判断。然而，一个更简单的方法是使用 AND( ) 和 OR( )逻辑函数。AND( )函数的用法是：

```
AND(logical test1, logical test2,...)
```

其中的参数 logical test1 和 logical test2 是逻辑测试。只有当所有的逻辑测试都成立时，AND( )函数才返回 TRUE，如果其中任意一个逻辑测试不成立，则返回 FALSE。

OR( )的用法与 AND( )的用法相似。只要其中任意一个逻辑测试成立，则 OR( )函数返回的值为 TRUE，如果全部逻辑测试都不成立，则 OR( )函数返回值为 FASLE。

在 A1:D6 单元里输入表 2-2 的数据，并把表格进行格式设置，得到如图 2-60 所示的电子表格。

	A	B	C	D
1	Part Number	Test 1	Test 2	Test 3
2	1	P	F	F
3	2	P	F	P
4	3	P	P	P
5	4	F	F	F
6	5	P	P	P

图 2-60

现在输入准则 1 的逻辑判断表达式。我们可以在 IF 语句里嵌入 AND( )函数。

在 E1 单元输入 Criterion 1(准则 1)。

在 E2 单元里输入公式“=IF(AND(B2="P",C2="P",D2="P"),"Accept","Reject")”，用填充柄把它复制到 E3:E6 单元里。

现在输入准则 2 的逻辑判断表达式。由于只需要通过其中一次测试，零件就符合标准，

因此我们要使用OR函数。

在F1单元里输入Criterion 2(准则2)。

接下来，在F2单元里输入“=IF(OR(B2="P",C2="P",D2="P"),"Accept","Reject")”，再用填充柄把它复制到F3:F6单元里。

经过格式设置后，我们得到一个如图2-61所示的电子表格。

	A	B	C	D	E	F
1	Part Number	Test 1	Test 2	Test 3	Criterion 1	Criterion 2
2	1	P	F	F	Reject	Accept
3	2	P	F	P	Reject	Accept
4	3	P	P	P	Accept	Accept
5	4	F	F	F	Reject	Reject
6	5	P	P	P	Accept	Accept

图2-61

2.6 教程：查找表的使用

在许多工程问题的实际求解中，我们要利用数据的表格特性。例如，在计算桥梁架构等结构的受力时，需要知道结构的材料以及材料的张力和弹性模量。在流体力学分析中，流体的密度和粘度很重要。许多教材用表格列出材料的这些属性，因此我们可以在这些表格里找到我们感兴趣材料的参数。Excel不但具有创建表格的能力，而且还可以为我们提供从表格中查找对应数据的能力。Excel用来查找表格中数据的函数包括VLOOKUP( )和HLOOKUP( )两个函数。

VLOOKUP( )函数可以查找表格中垂直排列的数据(因此首字符是V)，而HLOOKUP( )函数查找表格中水平排列的数据(因此首字符为H)。VLOOKUP( )的用法是：

```
VLOOKUP(lookup_value, table_array, col_index_num, range_lookup)
```

各个参数的作用如下：

- lookup_value为在表格第一列中查找的数值，它可以看成是列表数据的自变量。
- table_array为两列或多列数据，使用表格的单元区域。
- col_index_num为table_array中待返回的匹配值的列序号。col_index_num为1时，返回table_array第一列中的数值；col_index_num为2时，返回table_array第二列中的数值，以此类推。自变量，即与lookup_value相匹配的值在表格的第一列里。
- range_lookup是一个逻辑值，表示希望VLOOKUP查找精确的匹配值还是近似匹配值。
  - 如果为TRUE或省略，则返回精确匹配值或近似匹配值。也就是说，如果找不到精确匹配值，则返回小于lookup_value的最大数值。table_array第一列中的值必须以升序排序；否则VLOOKUP( )可能无法返回正确的值。
  - 如果为FALSE，VLOOKUP( )将只寻找精确匹配值。如果找不到精确匹配值，则返回#NA错误信息。

HLOOKUP( )的用法与 VLOOKUP( )一样，只是前者是相对行引用，后者相对于列引用。

例 2.7

水的沸点温度取决于水的压力。压力越高，水分子就靠得越近，则需要更多的动能才能够使水分子挣脱相互之间的束缚并转换为水蒸汽。分子的动能表现为水的温度，温度越高，动能越大。因此当我们提高水的压力时，水的沸点温度也随之提高。例如，我们知道，在一个大气标准压下，水的沸点为 212ºF。当大气压为两个标准大气压时，沸点的温度就提高到 248.8ºF。这样，我们可以说，当给定大气压力时，就可以求得水的沸点温度。反之给定水的沸点温度时，就可以求得大气压力。

在常压下我们把 1kg的水加到沸点所需要的能量叫做水汽化的热容量，用h_v表示。在常压下，把质量为m的水加热到沸点所需要的热量用方程(2.4)求出：

$$Q = mh_v \tag{2.4}$$

式中，Q的单位是kJ，m的单位是kg，h_v的单位为kJ/kg。

h_v的值与温度有关，经常用类似于表 2-3 的格式来表示。这个表格列出了沸点温度、压力以及在此温度和压力下水的汽化热容量。

表 2-3 水的汽化热容量

温度(°C)	压力(kPa)	水汽化热容量(kJ/kg)
0	0.6613	2501
25	3.169	2442
50	12.35	2383
75	38.58	2321
100	101.4	2257
125	232.1	2189
150	475.8	2114
175	892.0	2032
200	1554	1941
225	2548	1837
250	3973	1716
275	5942	1575
300	8581	1405
325	12 060	1190
350	16 510	893
374	22 090	0

以温度和质量为输入变量，建立一个电子表格，利用查找表确定h_v和压力，然后计算把水加热到沸腾所需要的热量。

解：

把表 2-3 的数据输入到 Excel 电子表格里，把列标题输入到 A1:C1 单元里。

右击工作表窗口底部的 Sheet1 表，选择【重命名】命令，如图 2-62 所示，然后输入 Table，作为表格名称。

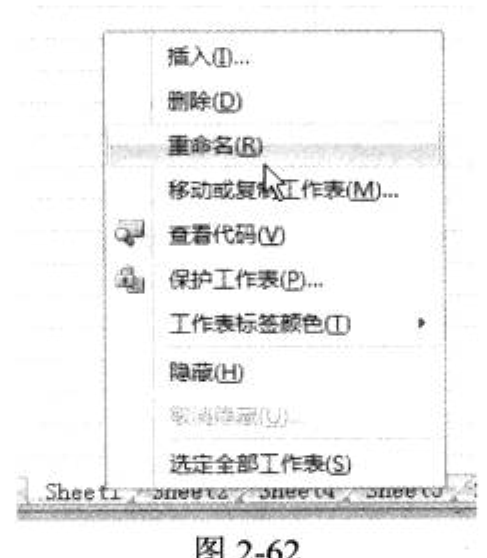


图 2-62

选择 Sheet2 表，就会打开第二个工作表，在这个工作表里，我们可以对隐藏在背后的表格进行计算。记住，有些表格非常大，因此，通常专门在另一个工作表进行数据分析，这会给我们带来很多便利。

把 Sheet2 重命名为 Calculations。

在 A1 单元里输入 Temperature(°C)，为了插入温度的符号，选择 Ribbon 的插入选项卡，然后在文本组里选择【符号】工具，如图 2-63 所示。

接着从符号对话框中找到温度的符号，如图 2-64 所示，单击【插入】按钮。

在 A2:A4 单元里分别输入 Mass (kg)，Pressure (kPa)和 Heat Required (kJ)。

在 B1 和 B2 单元里分别输入“25”和“3”。

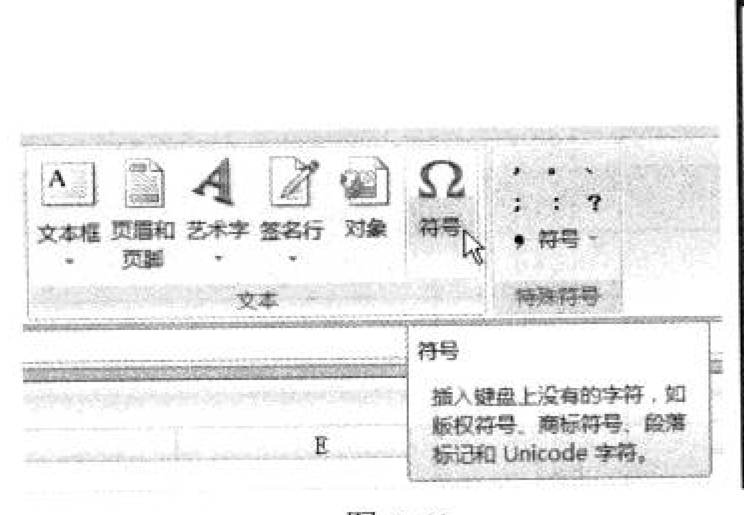


图 2-63

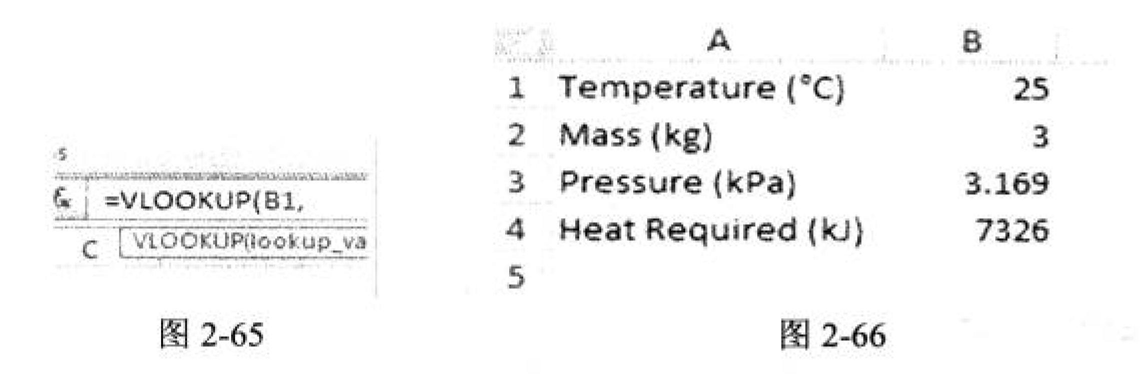


图 2-64

在 B3 单元里输入公式“**=VLOOKUP(B1,**”,这个公式下一个需要输入的参数是**table_array，或包含表格数据的单元区域。用鼠标点击是最容易插入单元区域的方法。公式编辑窗口仍处于活动状态，光标处在 table_array 参数位置，如图 2-65 所示。选择名为 Table 的工作表，选择 A2：C17 单元，然后输入“,2,TRUE)”，完成此公式的编辑。单击公式窗口的【输入】按钮，或者按 Enter 键。关闭公式编辑窗口。**

如果输入正确，则公式应该是“=VLOOKUP(B1,Table!A2:C17,2,TRUE)”。参数“Table!”表示引用本工作簿的另一个工作表里的单元区域。再次声明，用鼠标点击确定数据区域比起键盘输入公式要方便得多。

采用前面介绍过的鼠标点击法，在 B4 单元里输入公式“**=VLOOKUP(B1,Table!A2:C17,3, TRUE)*B2”，最后得到的计算结果应该如图 2-66 所示。**



	A	B
1	Temperature (°C)	25
2	Mass (kg)	3
3	Pressure (kPa)	3.169
4	Heat Required (kJ)	7326
5		

图 2-65　　　　图 2-66

分别输入温度(150)和质量(5)。读者得到计算结果应该类似于图 2-67。现在把温度改为 170，我们发现，计算结果并没有变化。这是因为，表格里并没有 170° C 的数据。因此 Excel 返回下一个最接近的温度值，即 150° C。如果需要更精确的结果，我们可以给表格添加更多的数据。

	A	B
1	Temperature (°C)	150
2	Mass (kg)	5
3	Pressure (kPa)	475.8
4	Heat Required (kJ)	10570
5		

图 2-67

现在我们给这个电子表格插入一个批注，提醒或警告用户，运算结果只是近似值。

右击 B1 单元，从弹出的快捷菜单中选择【插入批注】命令，如图 2-68 所示。

接着，在弹出的批注对话框里输入“Exact for tabulated temperatures(every 25ºC), else rounds down to next tabulated value.”(只得到与表格里温度(每隔 25° C)对应的结果，其他温度只能舍入到表格中最接近的值)，如图 2-69 所示。注意，在 B1 单元的右上角出现一个红色的三角形，表示该单元有一个批注。当我们把光标移动到 B1 单元时，就会出现这段批注内容。要编辑或删除这个批注，右击此单元，再选择合适的命令。

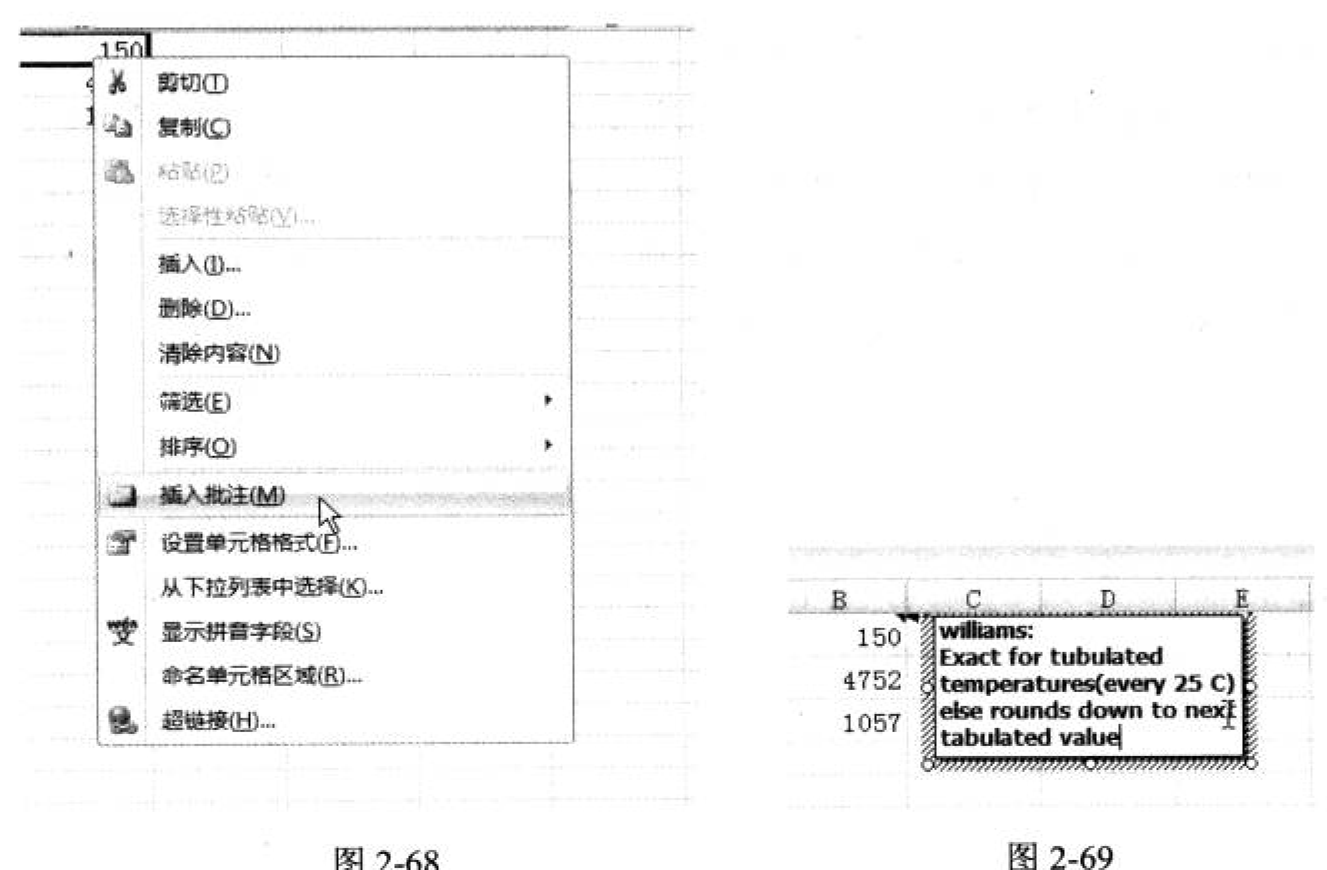


图 2-68　　　　图 2-69

2.7　教程：用 Excel 进行插值运算

查找表的缺点是，如果输入的值不是列表中的数据，则会把结果舍入到下一个最接近的值。用插值方法可以得到比较准确的结果。所谓插值就是估算两个已知点之间的数据。线性插值是最简单的插值方法。

例 2.8

某个城镇，每 10 年进行一次人口普查。普查的结果如图 2-70 所示。建立一个电子表格，利用线性插值方法，返回该城镇在 1900 年~2000 年之间每一年的人口估算值。

	A	B
1	YEAR	POPULATION
2	1956	
3		
4		
5	YEAR	POPULATION
6	1900	550
7	1910	720
8	1920	1256
9	1930	2287
10	1940	2756
11	1950	3012
12	1960	4086
13	1970	5220
14	1980	7478
15	1990	10112
16	2000	14226

图 2-70

解：

建立一个如图 2-70 所示的电子表格。

线性插值假定两组数据之间存在线性关系。考虑图 2-71，x_1和x_2表示两个自变量的值，或本例中的年份。相应的因变量的值用y_1，y_2表示，在本例里，它代表人口。用x代表x_1与x_2之间、且列表数据不存在的年份。y代表不在列表数据中的人口数量。假设人口与年之间存在一个线性关系。

方程(2.5)描述了因变量、自变量以及列表数据之间的线性关系：

$$y = y_1 + \frac{(y_2 - y_1)}{(x_2 - x_1)}(x - x_1) \tag{2.5}$$

因此，当我们要估计 1956 年的人口时，要执行以下运算：

$$y = 3012 + \frac{(4086 - 3012)}{(1960 - 1950)}(1956 - 1950) = 3656$$

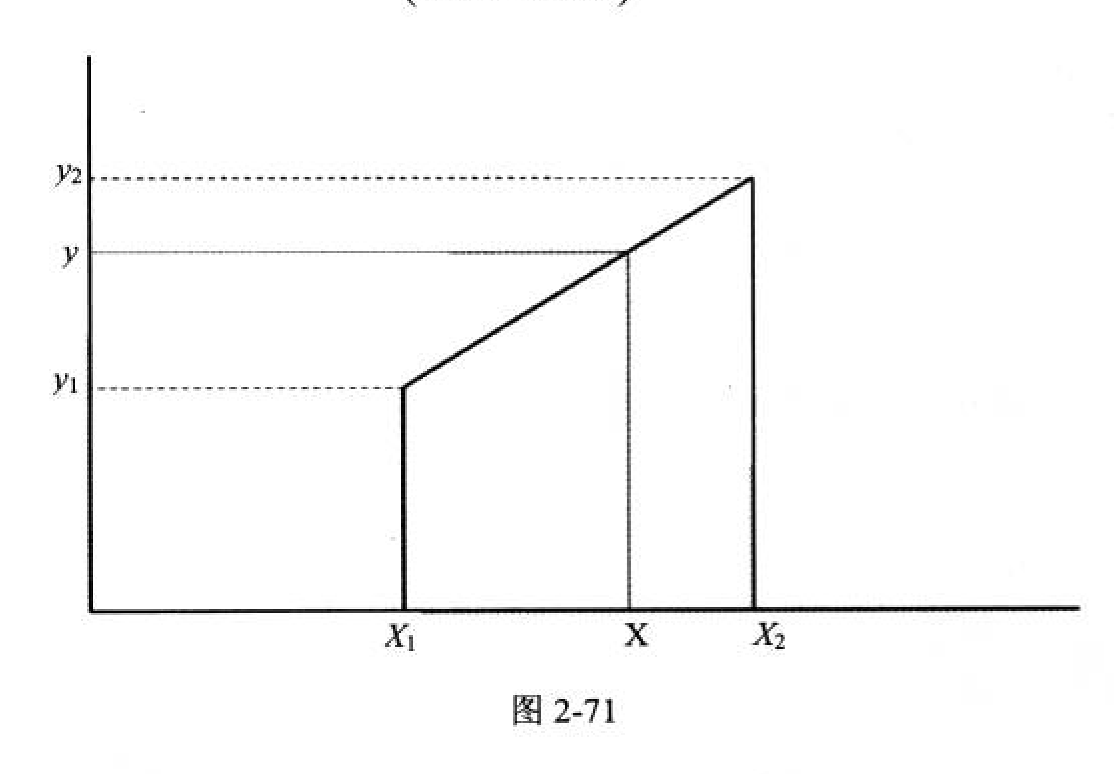


图 2-71

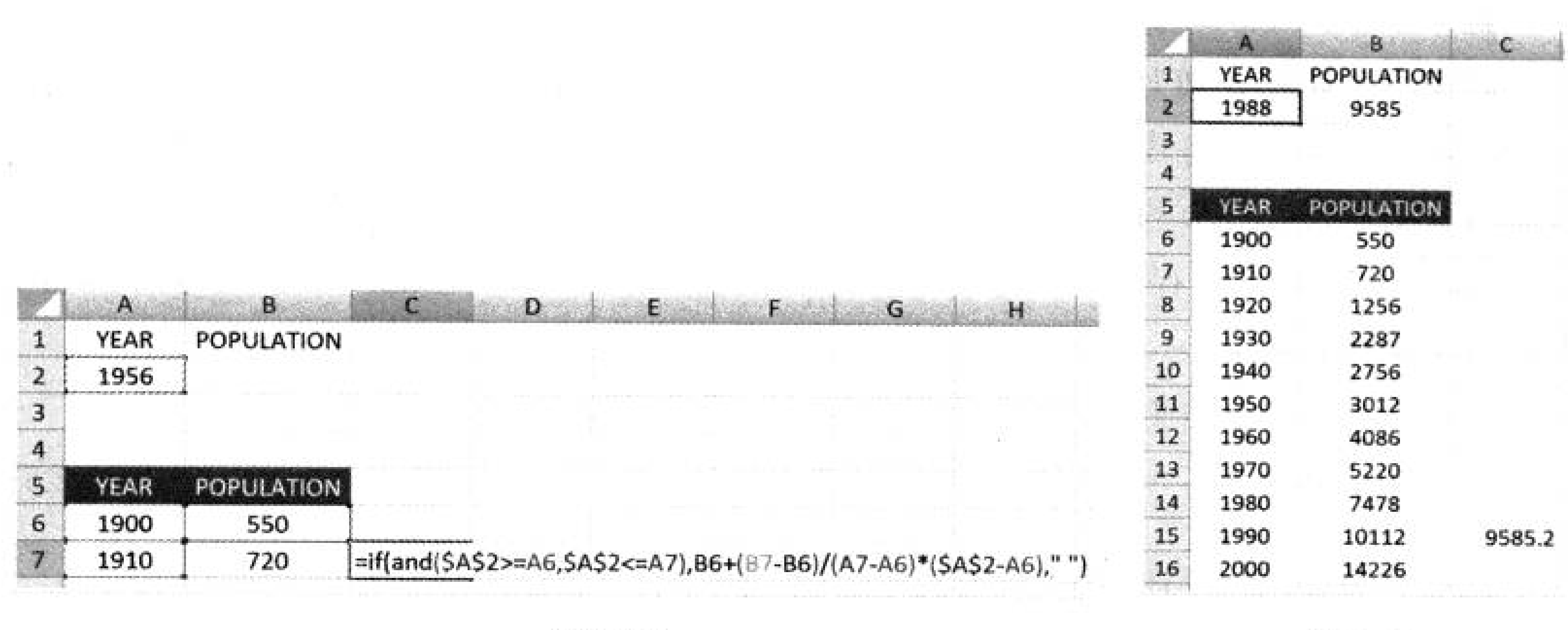
接着，我们要根据输入在 A2 单元里的年份，用逻辑判断选择插值运算需要用到的一组数据。

在 C7 单元里输入公式，如图 2-72 所示，并用填充柄向下复制到 C16 单元。

注意，在插值公式不起作用的单元里，就插入一个空格。实际上也可以插入一个 0 值，在我们这个例子里，插入 0 值，其结果并没有变化。然而，插入空格有两个好处。首先，只有求解值显示在 C 列，因此不会造成显示上的混乱。其次，通常一个表格可能是正值和负值，在没有作用的单元里插入空格，对于像 MAX( )这样的函数，仍然可以得到正确的答案，不管表格里包含的是正值还是负值。

根据插值结果，1956 年的估算人口出现在 C12 单元里。我们从 C 列向下扫描，提取这个值，把它放到 B2 里。我们还可以用 MAX(*range*)函数返回某个单元区域的最大值。

在 B2 单元里输入公式“=MAX(C7:C16)”，然后检查表格中的其他年份。最后生成的电子表格应该如图 2-73 所示。



	A	B	C	D	E	F	G	H
1	YEAR	POPULATION						
2	1956							
3								
4								
5	YEAR	POPULATION						
6	1900	550						
7	1910	720	=if(and($A$2>=A6,$A$2<=A7),B6+(B7-B6)/(A7-A6)*($A$2-A6)," ")					

图 2-72

	A	B	C
1	YEAR	POPULATION	
2	1988	9585	
3			
4			
5	YEAR	POPULATION	
6	1900	550	
7	1910	720	
8	1920	1256	
9	1930	2287	
10	1940	2756	
11	1950	3012	
12	1960	4086	
13	1970	5220	
14	1980	7478	
15	1990	10112	9585.2
16	2000	14226	

图 2-73

2.8 习题

1. 用 Excel 建立周历表(一周 5 天或 7 天)，安排一周的功课或活动。用不同的颜色表示不同的功课和活动。生成类似于图 2-74 的电子表格。

	Monday	Tuesday	Wednesday	Thurdsay	Friday
8:00-8:30					
8:30-9:00					
9:00-9:30		Graphics Flannigan 352			
9:30-10:00				Grpahics Flannigan 352	
10:00 - 10:30	Office Hours				
10:30-11:00					
11:00-11:30		Office Hours		Office Hours	
11:30-12:00					
12:00-12:30	Graphics Flannigan 312				
12:30-1:00			Grpahics Flannigan 312		
1:00-1:30	Special Topics				Office Hours
1:30-2:00					
2:00-2:30		COAD Flannigan 356			
2:30-3:00				Senior Design	
3:00-3:30					
3:30-4:00	Thermal Systems Austin 320		Thermal Systems Austin 320		Thermal Systems Austin 320
4:00-4:30					
4:30-5:00					

图 2-74

2. 表 2-4 显示了建立一个房子需要的任务、期限及各活动的依赖关系。

表 2-4 a 房屋的建造工序

工序	说明	紧前工序	期限(星期)
A	浇地基	无	1
B	建墙	A	2
C	安装房顶	B	3
D	安装水电设施	B	2
E	安装石膏灰胶纸夹板	C，D	2
F	安装门窗	C	1
G	安装地板	E，F	2
H	油漆	G	1

Excel 是可视化显示这些工序及它们之间的互相关系的很好的工具。工序的关系图也叫做甘特图或活动时间表。图 2-75 显示建造房屋活动的甘特图。在这个活动时间表里，我们用填充颜色表示工序的期限。有些工序，如 D 和 F，是在后续工序开始之前完成的，表示它是一项不受约束的工序，或者说该工序在工程进度中有松弛时间。具有松弛时间的工序即使耽误一段时间也不会影响整个工程的进度。如果某项工序时间上落后了就会影响整个工程的进度，这个工序就是关键路径工序。在这个项目里，关键工序是 A、B、C、E、G 和 H。

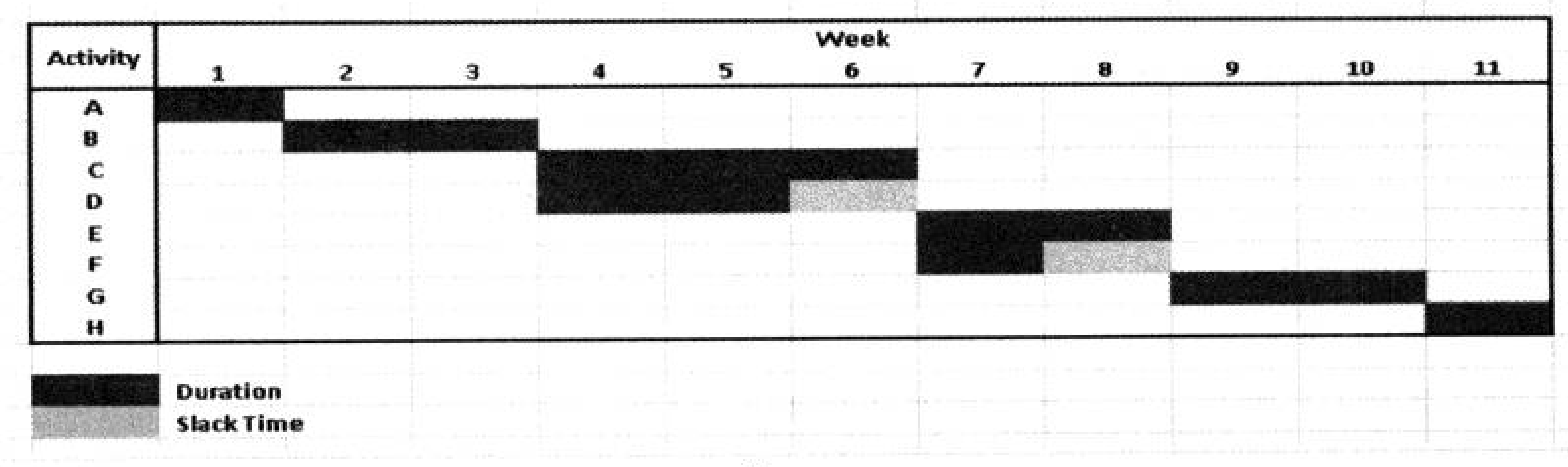


图 2-75

用 Excel 建立下表 2-5 中工序的甘特图。必须显示工序的期限和松弛时间。并求出关键路径工序。

表 2-5 工程项目的工序表

工序	描述	紧前工序	期限(天数)
A	决定探测速度		1
B	计算牵引力	A	1
C	决定发动机的马力和速度	B	1
D	决定滚筒的直径		1
E	决定滚筒的速度减速器	C,D	1

(续表)

工序	描述	紧前工序	期限(天数)
F	设计滚筒	D	2
G	集成回收系统	E,F	2
H	决定控制方案	C	1
I	选择控制元件	H	2
J	集成控制组件	I	3
K	编程 PLC	H	4
L	测试系统	G,J,K	2
M	完成文档	L	1

3. 灯泡是根据它消耗的功率(W)进行分级的。电能是按能量的单位出售的。它等于功率乘上时间。在美国，各地的电价相差很大。居民用电的价格大多在 0.06～0.23$/kW-hr。建立一个电子表格，它以电价和灯泡每天使用的时间为输入参数，然后计算 60W、75W 和 100W 白炽灯以及 15W 小型日光灯一年的费用。假设每个家庭使用 6 个 60W、4 个 75W 和 2 个 100W 的白炽灯灯泡，假设每天使用 6 个小时。计算一下，如果把所有的白炽灯换成日光灯，一年可以节省多少钱?

4. 建立一个电子表格，计算 1~100 整数的$\log_{10}$和$\log_e$(自然对数)的值。

5. 建立一个电子表格，计算 1~100 整数的平方根和立方根。

6. 建立一个电子表格，计算圆的直径从 1cm 增加到 100cm，步长为 1cm，所有圆的周长和面积。

7. 建立一个电子表格，计算所有边长为 1cm~100cm，步长为 1cm 的正方形的周长和面积。

8. 建立一个电子表格，计算所有边长为 1in~100in，步长为 1in 的立方体的表面积和体积。

9. 建立一个电子表格，计算所有半径为 1in~100in，步长为 1in 的球体的表面积和体积。

10. 建立一个电子表格，进行单位换算。输入为 SI 单位，输出为以下 US 自定义单位：

(a) 把 cm 转换为 in。

(b) 把°C 转换为°F。

(c) 把 N 转换为 bl。

(d) 把 m/s 转换为 mile/hr。

插入批注，说明单位换算引用的单元。

11. 建立一个电子表格，执行以下单位换算。输入 US 自定义单位的值，输出换算后以 SI 为单位的值。

(a) 把 ft 转换为米。

(b) 把lb/in²转换为N/m²(Pa)。

(c) 把 Hp 转换为 W。

(d) 把USgl转换为m³。

12. 分析第 1.1.2 节里的炮弹发射问题的解析解。建立一个电子表格，允许用户输入发射角度和初速度，用解析法计算最大高度、飞行时间和水平飞行距离。计算发射速度为150m/s，发射角为 60° 时炮弹飞行的最大高度、飞行时间和水平飞行距离。

13. 普通纸张的厚度为 0.004in。如果把这样的纸对叠一次，它的厚度就变成为 0.008in。再次对叠，它的厚度变成 0.016in。建立一个电子表格，显示纸对叠 0~50 次，它的厚度的变化。分别以 in、ft 和 m 为单位进行计算。

14. 我们从例 2.3 的淬火过程知道，在滚球轴承的制造过程中，各部件需要经过一道硬化工序。具体过程是先加热再迅速冷却，或者把它浸入到油槽或水槽里，这个过程即为淬火。滚球轴承的温度是时间的函数。$T(t)$可以按方程(2.3)估算：

$$T(t) = (T_i - T_\infty)e^{-t/\tau} + T_\infty$$

式中t为轴承浸入槽中的时间，单位为秒。T_i是轴承的初始温度，T_∞是油的温度。τ 是时间常数，单位是秒，它与轴承的材料、轴承的几何形状及油的特性有关。要求建立一个电子表格，以T_i、T_∞和τ 为输入变量，计算在 0~180 之间，步长为 1s的各时刻轴承的温度随时间的变化。假设时间常数τ =50s。假设轴承的初始温度为 800°C，油的温度为 40°C。把轴承的温度冷却到 100°C以下需要多长时间？

15. 表 2-6 是 5 位同学 7 次作业的分数。建立一个电子表格，求出每个同学的平均分，再根据 100 分制的等级规定给每个同学的等级(A，B，C，D，E)。100 分制等级规定是指90 分或 90 分以上为 A 等，小于 90 分大于等于 80 分为 B 等，其他的依此类推。

表 2-6　作业分数

	作业分数						
学生	1	2	3	4	5	6	7
A	95	60	89	90	92	80	87
B	92	100	93	87	90	85	90
C	88	60	76	89	70	40	60
D	90	87	70	89	92	85	85
E	78	90	94	89	98	95	97

16. 习题 2.14 的时间常数由以下方程确定：

$$\tau = \frac{\rho D c_p}{6h}$$

式中，p是轴承的材料密度，单位是g/cm^3。D是轴承的直径，单位是cm。c_p是材料的热容量，单位是J(g^{o}C)。h是油与轴承之间的热传导系数，单位是W/(cm^{2o}C)。表 2-7 列出铝、铜、金、镍和钢材料的热特性。

表 2-7　部分材料的热特性

	铝	铜	金	镍	钢
密度(g/cm^3)	2.77	2.77	19.30	8.90	7.83
热传导系数(J/(g°C))	0.875	0.875	0.129	0.444	0.434

假设热传导系数为 0.3W(cm^{2o}C)。建立一个电子表格，以上表中的材料密度D，热传导系数h、温度T_i和T_∞为输入变量。利用查找表，根据材料名称查找其热特性，然后计算从0~600s，步长为 1s的各个时刻的温度。假设初始温度为 400°C。轴承的直径为 0.5cm。并判断哪种材料冷却最快？

17. 建立一个电子表格，计算一个球的重量，单位为磅。输入是球的直径(单位为 in)。利用查找表，从图 2-76 的列表中查找某个材料的密度。利用图 2-76 核对球的计算公式。

	A	B	C	D	E	F
1	Diameter, inches	3.0				
2	Material	Aluminum		WEIGHT, lb	1.43	
3	Specific Weight, lb per cubic inch	0.101				
4						
5						
6		Aluminum	Steel	Brass	Titanium	Concrete
7	Specific Weight, lb per cubic inch	0.101	0.284	0.316	1.160	0.086
8						

图 2-76

18. 根据本章例 17，利用线性插值方法，不用查找表，计算水的沸点温度。

19. 在海平面，平均大气压为 14.7lb/in^2。在高原，压强会下降。随着压强的下降，在一定体积内的分子数也会随着下降。因此，棒球在Denver的Coor高地(海拔 5280ft)比在海平面上飞得更远。在珠穆朗玛峰(世界最高峰，海拔 29 035ft)，人们的呼吸将更加困难。建立一个电子表格，允许用户输入 0~50 000ft的海拔高度，根据表 2-8 估算大气压强。用线性插值法估算处于表中数据点之间的大气压。如果输入的值小于 0 或大于 50 000，在电子表格里将显示“超出范围”这样的错误信息。

表 2-8　大气压随高度的变化

高度(ft)	空气压力(lb/in^2)
0	14.70
1000	14.16
2000	13.66

(续表)

高度(ft)	空气压力(lb/in^2)
3000	13.17
4000	12.69
5000	12.23
10 000	10.10
20 000	6.76
30 000	4.37
40 000	2.73
50 000	1.69



第 3 章

MATLAB 基础

引言

MATLAB 是一个功能强大、应用广泛的专用计算工具。它应用在许多行业和研究单位。MATLAB 可以作为非常高级的计算工具，也可以作为低级的工程设计语言，或作为高级的图形编程语言(要利用 SIMULINK 接口)。它有数以千计的内置数学函数、图形函数和工程应用函数。它提供了各种特殊的工具箱，它些工具箱应用于一些专门的领域(如控制系统、信号处理、优化和模糊逻辑，等等)。

本章，我们将学习以下内容：

- MATLAB 环境的相关术语。
- 如何在交互模式下用 MATLAB 命令进行计算。
- 如何用 MATLAB 脚本自动进行运算。
- 了解 MATLAB 的脚本与函数的区别，掌握函数文件的开发和使用。

3.1 MATLAB 界面

单击如图 3-1 所示的 MATLAB 图标，可以启动 MATLAB 软件包。



图 3-1

图 3-2 是 MATLAB 启动后的界面。除了标准的 Windows 窗口工具栏外，在默认情况下还会出现另外 4 个窗口，它们是：

- 命令窗口：在这个窗口里可输入需要执行的计算命令。

- 当前目录窗口：显示当前目录里的文件列表，在 MATLAB 里被设置为当前工作目录。
- 工作区窗口：显示在当前会话过程中已定义的变量。
- 命令历史窗口：记录了在当前会话过程中所有已经执行的命令。

在默认情况下，命令窗口、当前目录窗口和命令历史窗口都是可见的。单击带标签的选项卡，可以隐藏或显示当前的目录窗口和工作区窗口。

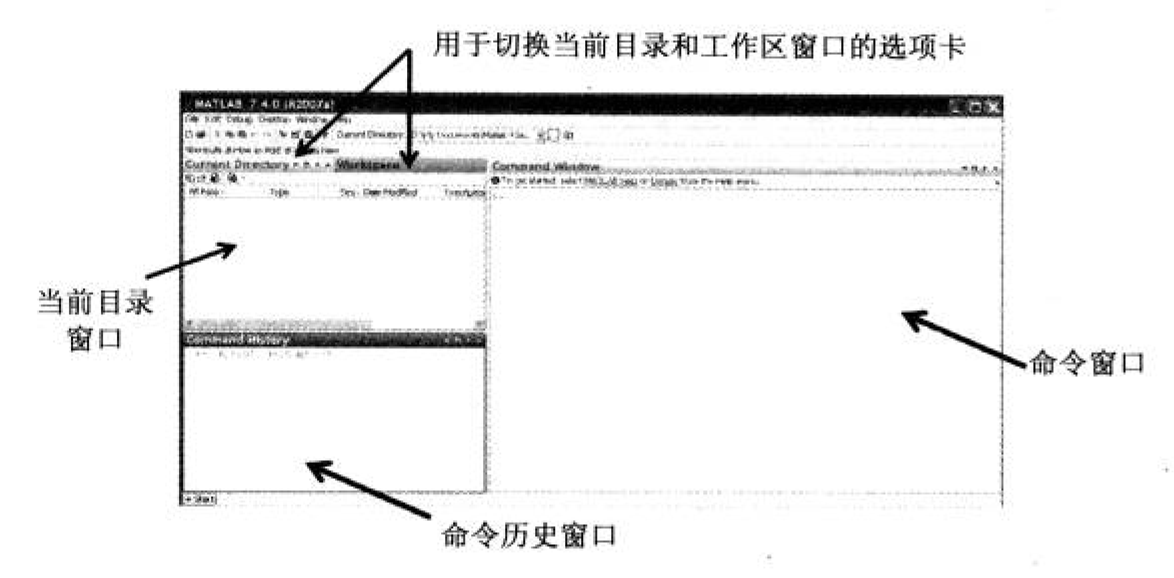


图 3-2

这种默认的显示方式很容易按个性化需要进行设置。选择 Windows 工具栏上的 Desktop(桌面)命令，可以自定义各个 MATLAB 窗口的关闭、最小化或打开方式，如图 3-3 所示。

本章主要使用命令窗口和工作区窗口。

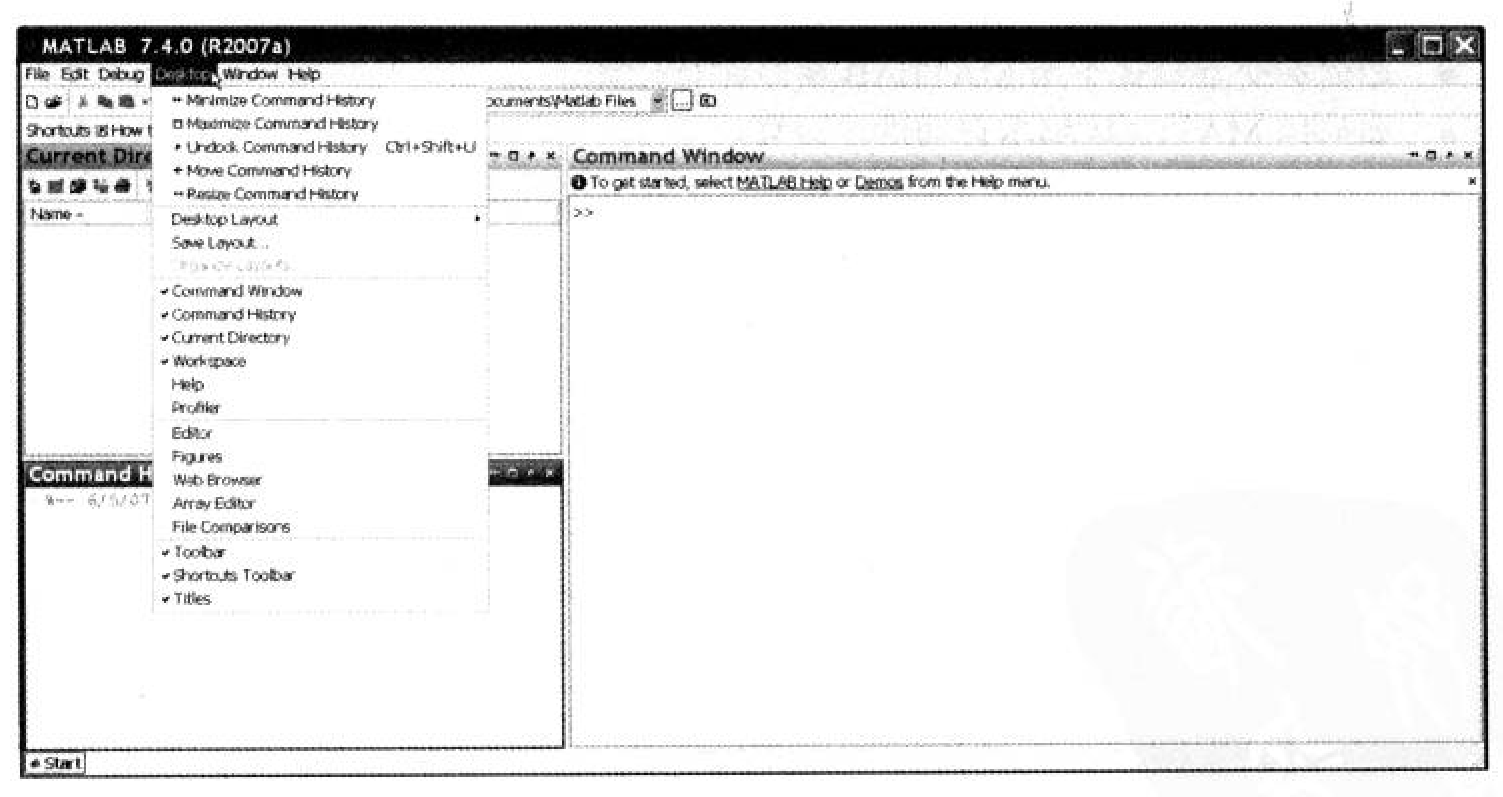


图 3-3

3.2 教程：利用命令窗口进行交互计算

现在我们开始介绍将命令窗口作为交互计算模式的用法。在这个模式中，我们把MATLAB当作一个科学计算器。在本教程里，我们将用这种模式求解一些几何和三角函数问题。

例 3.1

一个矩形的底边为7.500in，高为3.300in，我们用MATLAB求它的面积。

解：

根据几何知识，我们知道，矩形的面积a是底边b与高h的乘积，即：

$$a = bh \tag{3.1}$$

单击命令窗口，“>>”符号是命令提示符，也是命令开始输入的位置。在命令窗口的提示符后输入如图3-4所示的命令。

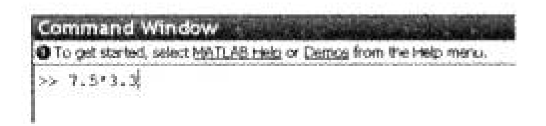


图 3-4

星号(即按下Shift+8组合键)是乘法运算符。按下Enter键，就得到了这个运算符两边数字相乘的结果：

```
>> 7.5*3.3
ans =
24.75
```

根据显示结果。我们得出结论，这个矩形的面积是24.75in^2。单击工作区选项页，显示工作区窗口，如图3-5所示。

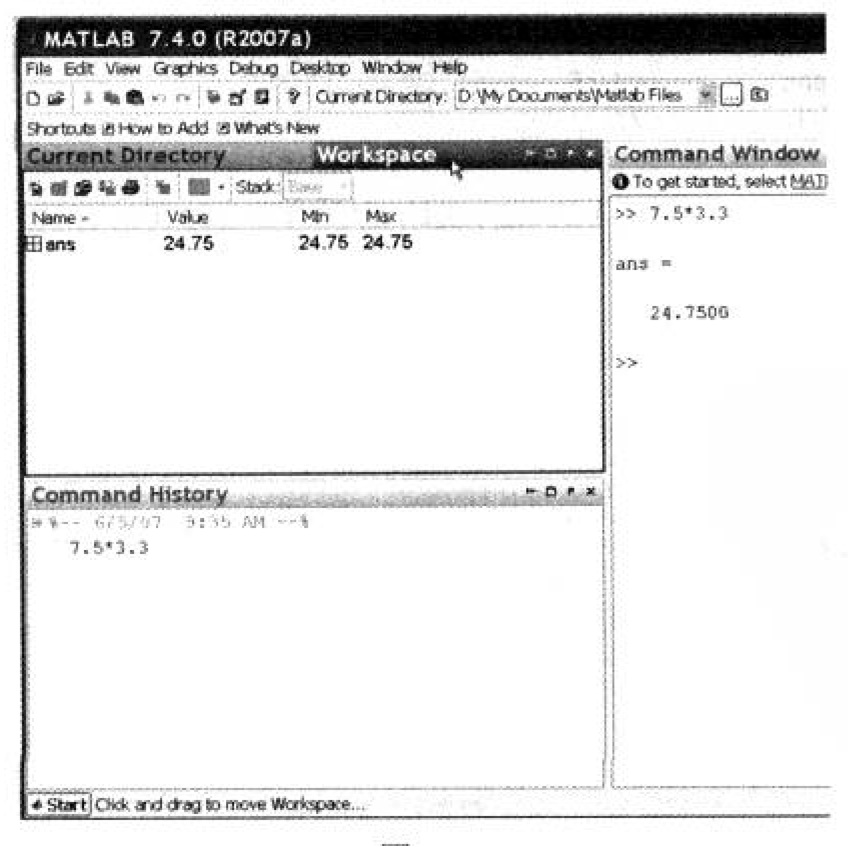


图 3-5

需要注意的是，命令窗口用一个名为 *ans* 的变量显示运算结果。从工作区窗口里可以看出，新增了一个变量 *ans*，并显示保存变量 *ans* 里的值是 24.75。另外还要注意的是，命令历史窗口也得到更新。我们会在列表中看到刚才在命令窗口输入的命令。

面积的另一种计算方法

在这个计算面积的问题里，我们只是把一个需要计算的算术表达式输入到命令窗口中。将运算结果保存到 *ans* 变量里。当 MATLAB 执行数值运算时，*ans* 是它的默认变量名。为了进行比较复杂的计算，也为了实现 MATLAB 作为一个编程语言的强大功能，我们需要利用变量比较复杂的用法。在另外一种求解方法中，我们将进一步深入研究变量在计算环境下的建立方法。

变量是一个由字母和数字混合组成的符号。它可以被赋予一个数值(类似于一个代数方程)。在 MATLAB 里，一个变量名必须是以 26 个字母之一开始，不能有空格(或某个特殊字符)，严格区分大小写。不能与 MATLAB 预先定义的命令名称同名。为了避免出现与变量有关的问题，这里，我们使用非常简单的变量名。

我们先建立变量 *b*1，并把矩形的底边长赋给它。为此我们在命令窗口执行以下命令：

```
>> b1=7.5
```

按 Enter 键，执行这个命令，系统就建立了一个名为 *b*1 的变量，并把 7.5 赋给这个变量。结果回显到屏幕上：

```
>> b1=7.5
b1 =
    7.5000
```

注意，现在变量 *b*1 已被添加到工作区窗口里，如图 3-6 所示。

Current Directory　Workspace

Stack: Base

Name	Value	Min	Max
ans	24.75	24.75	24.75
b1	7.5	7.5	7.5

图 3-6

刚才我们使用了赋值运算。“=”就是赋值运算符。虽然与“等号”相似，但是它的作用完全不同。在 MATLAB 里，赋值运算符的作用相当于“把赋值号右边的数赋给赋值号左边的变量”。正确理解赋值号的用法是掌握 MATLAB 编程的关键。许多刚开始学编程的学生总是搞不清赋值号与数学中的等号的区别。记住，从数学上讲“*b*1=7.5”与“7.5=*b*1”是等效的，但是在 MATLAB 程序里，它们完全不同。只有前一个表达式才可以理解为赋值运算。

现在我们将定义第二个变量保存矩形的高度。定义一个变量 *h*1，并用赋值号把矩形的高度赋给它：

```
>> h1=3.3
```

同样，按 Enter 键，执行这个赋值运算。结果将回显在屏幕上。更新工作区窗口，把这个新变量添加到列表里。

现在我们要建立一个公式计算矩形的面积。要用到前面定义的变量和乘法运算符，赋值运算符将把运算结果赋给一个变量。在命令窗口里输入以下命令：

```
>> a=b1*h1
```

按 Enter 键，MATLAB 就执行乘法运算(把变量的当前值代入方程里)，并把运算结果赋给变量 a。

```
>> a=b1*h1
a =
    24.7500
```

工作区的内容将会同时得到更新。

在这个例子里，我们引入了两个运算符：乘法运算符和赋值运算符。与第 2 章里的 Excel 一样，MATLAB 还有很多其他数学运算符。表 3-1 列出了一些最常用运算符。

表 3-1 数学运算符

运　　算	运 算 符	MATLAB 里的用法
加法	+	>>a=6.3+2.2
减法	-	>>c=10.2-4.5
乘法	*	>>f=3.4*2.7
除法	/	>>k=3/7.3
指数	^	>>p=4.2^2

例 3.2

用 MATLAB 工具求例 3.1 的矩形周长(底边 7.5in，高 3.300in)。

解：

根据几何知识，矩形的周长可以由以下公式计算：

$$p=2b+2h \tag{3.2}$$

回忆一下，这里的底和高的长度与例 3.1 相同。由于我们还没有关闭 MATLAB，也没有清除内存变量，因此底和高的变量还在工作区里，在计算周长时我们显然可以重用这些变量。在命令窗口输入以下表达式：

```
>>p=(2*b1)+(2*h1)
```

计算结果是 21.60in，并把这个值赋给变量 p。注意，这里的括号是用来说明运算的优先次序。在本例中，我们希望先执行乘法运算，再执行加法运算。还记得吗？我们在第 2 章曾说过，包括 MATLAB 在内的大多数计算软件，用一种标准的层级次序解释数学运算的优先次序。与 Excel 一样，MATLAB 的数学运算优先规则是：

- 运算次序先左后右。
- 指数运算的优先级最高。

- 乘法运算和除法运算优先级其次。
- 加法和减法运算优先级最低。

使用括号可以改变这种标准的运算次序。当表达式含有括号时，先执行最里层括号里的表达式，从里到外，最后计算最外层的括号。一个好的习惯是，为了使表达式按我们的愿望执行，尽量多用一些括号。当用计算工具得到的计算结果与预期结果(来自手工计算的结果或其他方法得到的结果)不一致时，往往错误的原因就出于运算符优先级的次序。

注意，在我们这个例子里，括号确实没有必要，根据数学运算优先级规定，乘法运算优先级高于加法运算符。因此，即使不用括号，这个表达式也能得到正确的结果。许多工程技术人员很乐意使用括号，因为利用括号可以在外表上帮助我们“组织”方程，能清楚表明程序员的意图。

例 3.3

一个直角三角形如图 3-7 所示。用 MATLAB 求 x 和 y 的值。

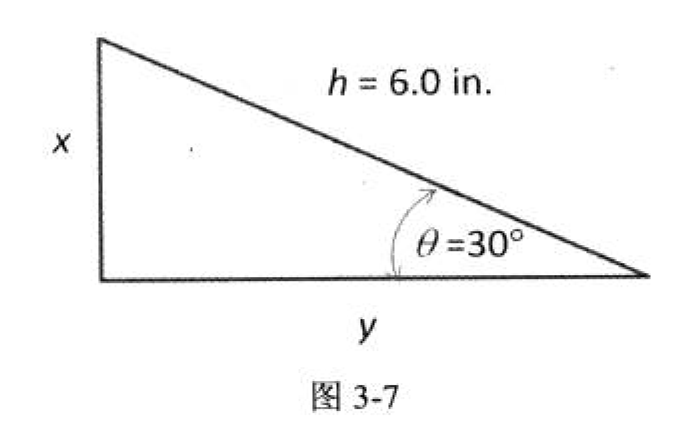


图 3-7

解：

在例 3.1 和例 3.2 中，我们只用了最基本的运算符解决一些最简单的问题。本例要用到三角函数，我们还记得以下的公式：

$$x = h\ \sin\theta \tag{3.3}$$

$$y = h\cos\theta \tag{3.4}$$

既然已经完成了有关矩形的计算，那么就可以用下面的命令复位工作区(即清除工作区里的全部变量)：

```
>> clear
```

现在，工作区中没有变量了。此外还可以用下面的命令复位命令窗口：

```
>> clc
```

输入下面的命令并按 Enter 键，可以定义一个变量(用于保存三角形的斜边值)：

```
>> hyp=6;
```

注意表达式后面的分号。分号的作用是抑制命令执行时将结果“回显”在屏幕上。注意，这个命令将在工作区中建立 HYP 变量，但是它的值并没有显示在命令窗口。通常，当我们定义变量时，没有必要在命令窗口显示它的值，因此要使用分号。下面我们在表达式后面将省略分号，因为我们只需要看到这些表达式的运算结果。

要求出 x 的值，应输入以下的表达式：

```
>> x=hyp*sin(30*pi/180)
```

如果在表达式后面省略了分号，则当我们按 Enter 键时，计算结果就会显示在命令窗口上：

```
>> x=hyp*sin(30*pi/180)
 x =
   3.0000
```

即 x=3.0 英寸。

在这个表达式里需要注意几件事。第一个是 *sin* 函数的用法。它是 MATLAB 平台众多内置的、用于复杂计算的函数之一。通常，MATLAB 里的一个函数需要在其后的括号里提供一个或多个数(称为函数的参数)，并返回一个计算结果。在 MATLAB 里，函数的一般语法是函数名及用括号表示的参数，以后我们会经常用到。要浏览 MATLAB 里包括三角函数、指数函数和其他复杂函数的全部数学函数列表，只需要使用在线帮助文件。任何时候输入以下命令就可以访问这些在线帮助文件：

```
>>help
```

输入这个命令后，MATLAB 就会列出多种可以访问的帮助主题。要看初等函数，可以从帮助主题列表中选择 MATLAB\elfun 超链接，或者输入以下命令：

```
>> help matlab\elfun
```

则会在命令窗口显示函数列表。需要用滚动条才能看到整个列表。

当我们分析 sin 函数时，必须注意另一个重要的事情，即 sin 函数的角度参数必须以弧度，而不是度为单位。在这个表达式里的括号中，我们直接把度转换为弧度。通常，工程技术人员使用的大多数编程语言和计算工具都使用弧度作为三角计算的基本单位。这可以从 sin 函数的帮助文档中得到证实，从 elfun 列表中选择合适的超链接，或者在命令窗口输入以下命令：

```
>> help sin
```

帮助文档说明了 sin 函数要求的角度参数必须以弧度为单位。对于刚开始学习编程的程序员和工程专业的学生，在工程计算工具中，角度参数要用弧度而非度是最常被忽略的问题。注意，MATLAB 确实还提供了另一组三角函数，它们以度为单位。例如，sind 是 sin 函数的另一种形式，但它的参数要以度为单位。通过在线帮助文档可以证实这一点：

```
>>help sind
```

此外，在 sin 函数的参数单位转换表达式里，我们肯定会注意到另一件事：那就是变量 pi 的使用。由于 π 这个值在工程计算中会被频繁用到，因此 MATLAB 专门为我们定义

了这个变量 pi。虽然它并没有出现在工作区窗口里，但是我们随时都可以使用这个变量。请读者注意，千万不要在自己的程序中再定义这个变量！在 MATLAB 里我们可以覆盖这个变量的默认值，因此不要在赋值号的左侧使用 pi 这个变量。

本例子的最后，我们要求出 y 的值，为此输入以下的命令：

```
>> y=hyp*cos(30*pi/180)
```

或者：

```
>> y=hyp*cosd(30)
```

不管采用哪个式子，输出结果都是：

```
y =
     5.1962
```

我们需要注意的是，虽然 MATLAB 在计算中使用较多的小数位数，但在默认情况下，MATLAB 的输出结果使用短格式，即只显示 4 位小数。用下面的命令，可以显示更多的小数位数：

```
>> format long
```

设置了长格式后，重新计算得到的结果是：

```
>> y =hyp*cosd(30)
y =
   5.196152422706632
```

用下面的短格式命令，恢复到原来的默认格式：

```
>> format short
```

不管显示多少位小数位数，计算的精度是由输入数据的精度决定的。我们可以把上面的结果理解为 5.2in。

在继续本例之前，我们可以用多种方法验证计算结果。一种方法是，根据前面计算得到的 x 和 y 值，能否得到输入角度 30°，为此，输入以下命令：

```
>> theta = atand(x/y)
```

这个命令用于计算角度的正切值，单位是度。结果确实是输入值 30°。

此外我们还可以利用勾股定理验证斜边的长度，MATLAB 提供了求平方根函数 sqrt( )：

```
>> h = sqrt(x^2+y^2)
```

注意，这个表达式计算三角形两条边的平方的和，再把这个和作为平方根函数的参数。结果是 6.0in。与输入值相符。

平方根函数是另一个初等函数。MATLAB\elfun 目录提供了一组常用的初等数学函数，我们把其中一部分列在表 3-2 里。

表 3-2 一些常用的初等函数

函 数	MATLAB 里的用法	数 学 意 义
自然对数	log(x)	$\ln(x)$
以 10 为底的对数	log10(x)	$\log(x)$
指数	exp(x)	e^x
绝对值	abs(x)	$\|x\|$
平方根	sqrt(x)	$\sqrt{x}$
四舍五入	round(x)	
正弦函数(弧度)	sin(x)	$\sin(x)$
正弦函数(角度)	sind(x)	
余弦函数(弧度)	cos(x)	$\cos(x)$
余弦函数(角度)	cosd(x)	
正切函数(弧度)	tan(x)	$\tan(x)$
正切函数(角度)	tand(x)	
反正弦函数(弧度)	asin(x)	$\sin^{-1}(x)$
反正弦函数(角度)	asind(x)	
反余弦函数(弧度)	acos(x)	$\cos^{-1}(x)$
反余弦函数(角度)	acosd(x)	
反正切函数(弧度)	atan(x)	$\tan^{-1}(x)$
反正切函数(角度)	atand(x)	
反正切函数(弧度,四象限)	atan2(y,x)	

3.3 教程：MATLAB 脚本文件的使用

虽然 MATLAB 的交互计算模式是一个功能强大的工具，但是当我们开始用编程实现自动计算时它的强大之处才能显示出来。本节将介绍脚本文件的作用。我们在命令窗口执行的任何运算都可以保存在脚本文件里自动执行。首先离线建立脚本文件，然后在命令窗口执行这个文件。可以反复多次执行这个脚本文件，这样，我们可以执行一些重复性的计算，而无需每次重新输入命令。

现在设计一个脚本文件，它可以自动实现前面例 3.1 和例 3.2 里的面积和周长计算。在开始编写脚本之前，我们必须设置一个目录，专门用来保存 MATLAB 脚本文件。在窗口的工具栏，显示当前目录的位置，选择【...】图标，弹出 Browse For Folder(文件夹浏览)对话框，如图 3-8 所示。

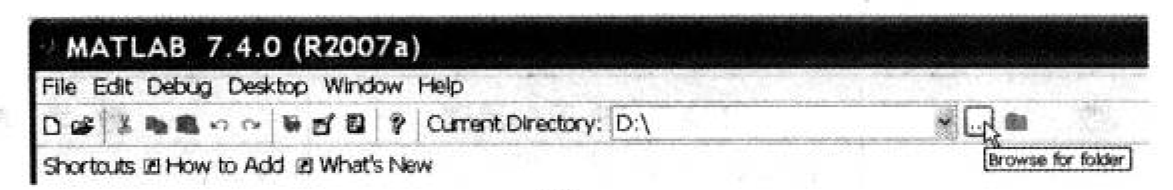


图 3-8

找到我们打算用来保存脚本文件的目录，单击如图 3-9 所示的 Make New Folder(建新文件夹)按钮。



图 3-9

输入文件夹的名称，并按 OK 键。现在我们要把新建立的文件夹作为当前目录。这样，我们在 MATLAB 会话过程中执行的一切计算都将保存在这个目录里。如果我们要把这个目录设置为 MATLAB 默认的当前目录，则必须要把它添加到 MATLAB 的路径列表(一个与 MATLAB 工作环境有关的目录树)。要把我们的目录添加到路径里，可选择 File:SetPath…命令，如图 3-10 所示。

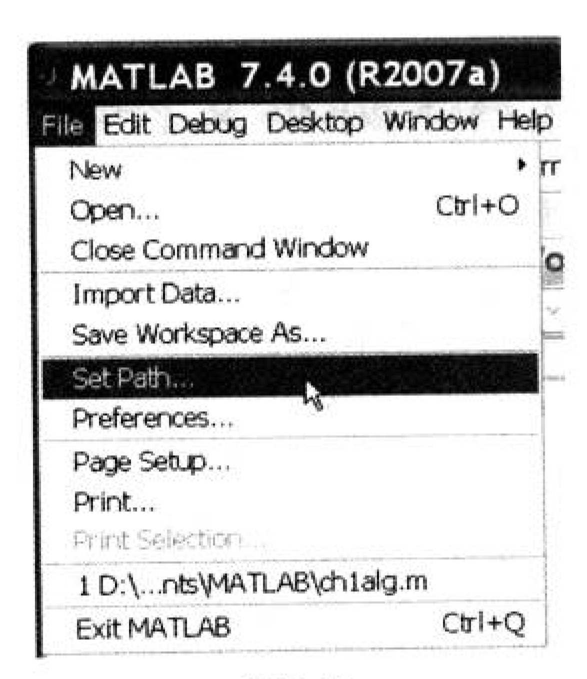


图 3-10

出现 Set Path(设置路径)对话框，单击 Add Folder(添加文件夹)按钮，如图 3-11 所示。从 Browse For Folder(文件夹浏览)对话框中找到我们刚建立的目录。再单击 OK 按钮，

如图 3-12 所示。

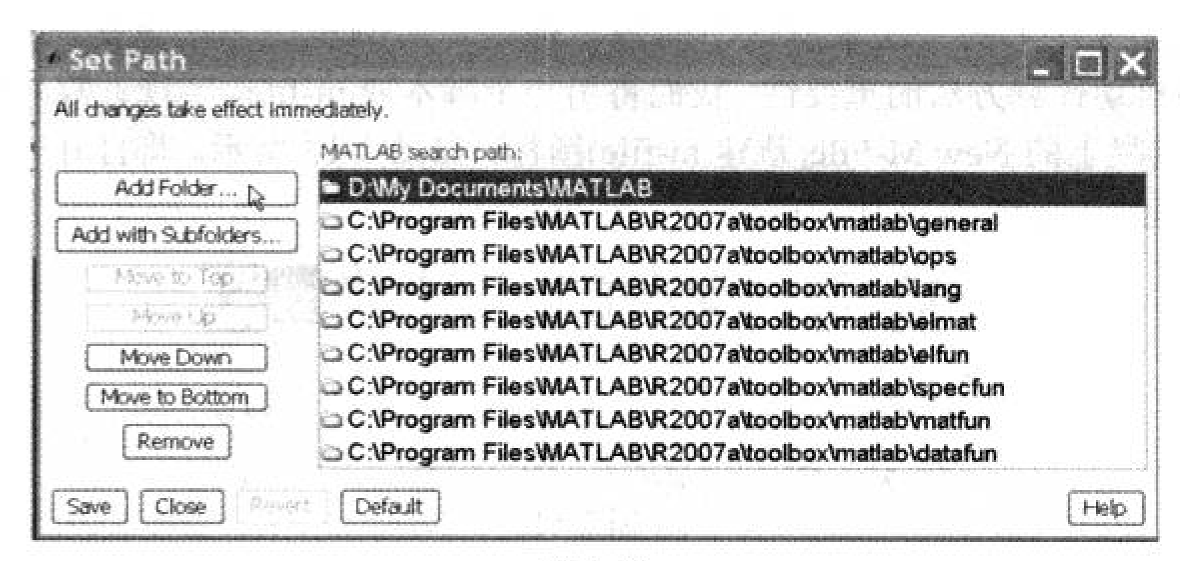


图 3-11

图 3-12

现在这个目录已经被添加到 MATLAB 的路径里。这非常重要，因为当 MATLAB 搜索文件时，它只会对 MATLAB 路径里的目录进行搜索。单击路径设置对话框的 Save(保存)按钮，保存新路径，并关闭对话框。为了更好地组织自己的 MATLAB 生成的文件，读者可能还想建立更多的工作目录。例如，读者可能想把 MATLAB 的每一节课的内容保存到一个单独的目录里。注意，每个新建立的目录都必须添加到 MATLAB 的路径里。

例 3.4

计算表 3-3 里的 3 个矩形的面积。

表 3-3　矩形尺寸

	底边	高
矩形 1	6.5in	2.1in
矩形 2	7.2in	3.0in
矩形 3	7.5in	3.3in

解：

可以用例 3.1 中的方法求解这个问题。但是，本例里的多次重复运算正好表现了 MATLAB 的自动计算方法的重要性。我们将用一个脚本或 m-file，实现计算过程的自动化。

单击工具栏上的 New M-File(新建 m-file)图标，如图 3-13 所示。将打开一个编辑窗口，在编辑窗口里输入如图 3-14 所示的脚本，用 Enter 键换行。



图 3-13

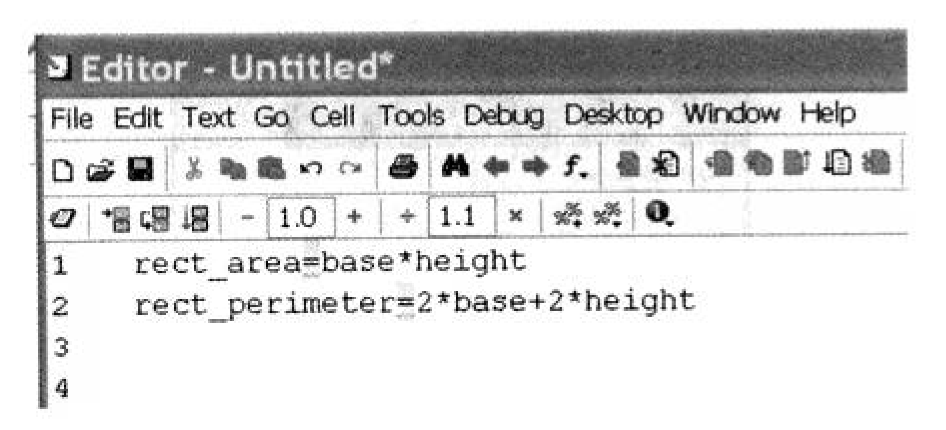


图 3-14

注意，脚本中的表达式与例 3.1 中使用的表达式非常相似(尽管，两者使用了不同的变量名，但这只是为了演示目的)。从编辑菜单选择 File(文件) | Save(保存)命令，如图 3-15 所示。

脚本文件保存的位置就是前面我们设置的工作目录。输入文件名 rect(在默认情况下，它的扩展名为.m)，再按 Save(保存)按钮，如图 3-16 所示。



图 3-15

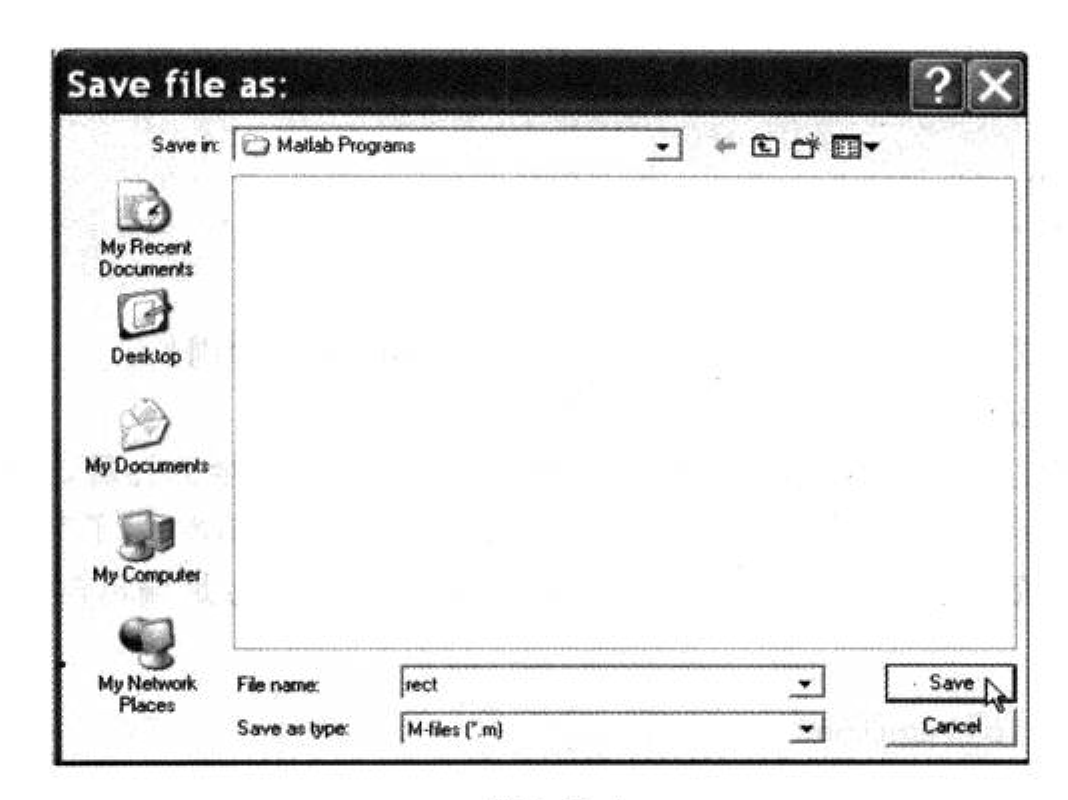


图 3-16

关闭编辑窗口，回到命令窗口。在命令窗口输入 clear 和 lc 命令，清除工作区和命令窗口。

在执行脚本之前，我们必须定义 *base* 和 *height* 变量。在命令窗口输入下面的命令，输入矩形 1 的底边和高。

```
>> base=6.5;
```

和：

```
>> height=2.1;
```

还记得命令后面的分号的作用吗？它抑制了将命令的结果回显在屏幕上。注意，现在工作区里已增加了 *base* 和 *height* 变量。

定义了这些变量之后，我们执行这个脚本，计算矩形的面积和周长。执行脚本只需要在命令窗口输入脚本文件名：

```
>> rect
```

按 Enter 键就会立刻执行这个脚本。就如同在命令窗口里输入公式那样，MATLAB 解释这个脚本程序。由于脚本里每个命令后面都没有分号，因此运算结果将回显在屏幕上。

```
>>rect
rect_area =
 13.6500
rect_perim =
 17.2000
```

对于其他两个矩形，我们只需要在命令窗口给 *base* 和 *height* 赋新矩形的值，可以再次执行 rect 脚本：

```
>> base =7.2;
```

```
>> height=3.0;
>> rect
rect_area =
    21.6000
rect_perim =
    20.4000
```

作为练习，读者可以重复此过程，求第三个矩形的面积和周长。

另外一个办法

我们可以采用另一种办法，让脚本自动执行三个矩形的全部计算。选择 File(文件) | Open(打开)命令，重新打开 rect 脚本，根据下面的内容修改脚本(为了节省键盘输入操作，也可以使用传统的复制/粘贴的方法，注意，每行前面的行号是编辑窗口自动添加的，不必输入)：

```
1  % Define dimensions of Rectangle 1
2  base1=6.5;
3  height1=2.1;
4  % Define dimensions of Rectangle 2
5  base2=7.2;
6  height2=3;
7  % Define dimensions of Rectangle 3
8  base3=7.5;
9  height3=3.3;
10 %Compute values for Rectangle 1
11 rect_area1=base1*height1
12 rect_perimeter1=2*base1+2*height1
13 %Compute values for Rectangle 2
14 rect_area2=base2*height2
15 rect_perimeter2=2*base2+2*height2
16 %Compute values for Rectangle 3
17 rect_area3=base3*height3
18 rect_perimeter3=2*base3+2*height3
```

我们注意到，1、4、7、10、13 和 16 行的行首是百分号符号("%")，脚本中这些行是注释行。MATLAB 不会对这些行进行解释，它们只是为了说明脚本中各语句的作用。虽然注释语句并不会影响程序的执行，但是对其他人理解本程序会大有帮助。或者在将来某个时候，继续前面的工作时，注释会告诉我们脚本的功能。读者必须养成在程序中多加注释的好习惯。

在前面的例子里，运行脚本的方法是先保存脚本，然后在命令窗口的提示符后输入脚本文件名。但是在这个例子里，我们用另外一种方法执行这个脚本，即窗口容器的快捷命令。单击编辑工具栏上的 Save and Run(保存并运行)图标，如图 3-17 所示。



图 3-17

这个快捷命令将保存我们对脚本文件所做的修改，并且在命令窗口执行这个脚本文件。这个脚本在命令窗口的输出结果如下所示：

```
rect_area1 =
    13.6500
rect_perimeter1 =
    17.2000
rect_area2 =
    21.6000
rect_perimeter2 =
    20.4000
rect_area3 =
    24.7500
rect_perimeter3 =
    21.6000
```

仔细分析输出结果，我们将看出，命令行后分号的作用。由于定义底边/高的命令后带有分号，因此它们的结果并没有回显在屏幕上。而计算面积和周长的命令后面没有分号，因此，计算结果显示在命令窗口里。利用这种简单的办法，输出用户需要的结果。

例 3.5

一个用户希望能够计算任何矩形的面积和周长。要求设计一个脚本实现此功能。

解：

我们要设计一个交互的脚本程序，它提示用户输入矩形的底和高，并且输出面积和周长。为此要用到 MATLAB 里一个名为 input( )的函数。该函数在屏幕上显示一个信息，并等待用户输入响应。当用户按 Enter 键后，就把用户的响应赋给某个变量。例如，命令：

y=input('输入 1 到 10 之间一个数');

将显示单引号内的提示信息，并把用户输入的数据赋给变量 *y*。在之后的计算中，可以引用变量 *y* 的值。(读者想进一步了解 input( )函数的用法，在命令窗口输入 help input，阅读在线帮助文档。)

新建一个 m-file，输入以下的脚本程序：

```
1 %Get user input
2 b1=input('Enter the base value (in inches)');
3 h1=input('Enter the height value (in inches)');
4 %Compute area
5 input('Hit the Enter key to compute area');
6 area=b1*h1
7 %Compute perimeter
8 input('Hit the Enter key to compute perimeter');
9 perim=(2*b1)+(2*h1)
```

把这个脚本文件保存为 rect2，注意脚本中的以下内容：

- 第 2 行和第 3 行使用 input( )函数，让用户输入矩形的底和高的数值，并把它们赋给变量 *b*1 和 *h*1。

- 第 5 行和第 8 行实质上是“哑”输入命令，它会暂停执行之后的命令，直到用户按下 Enter 键为止。注意，这里的输入值并没有赋给某个变量，因此输入值无法得到利用。这类“哑”输入只是为了方便用户操作，并非必要的。

在命令窗口输入脚本文件名或者通过快捷图标，执行此脚本程序。在命令窗口中，脚本将显示第一个输入提示信息，并等待用户输入：

```
Enter the base value(in inches)
```

输入 6.5 并按下 Enter 键，脚本显示第二个输入提示信息，并再次等待用户输入：

```
Enter the base value (in inches)6.5
Enter the height value (in inches)
```

输入 2.1 并按下 Enter 键。连续两次按下 Enter 键，响应后面两个的等待请求，MATLAB 将计算并输出面积和周长：

```
Enter the base value (in inches)6.5
Enter the height value (in inches)2.1
Hit the Enter key to compute area
area =
   13.6500
Hit the Enter key to compute perimeter
perim =
   17.2000
```

自从上次清除工作区以来定义的每个变量都将显示在工作区窗口里。这些变量将在后面的计算和脚本中用到。我们称这些变量为全局变量。在我们用 clear 命令清除这些命令，或者关闭 MATLAB 程序之前，这些变量和它们的值将一直存在。

在这一节里，我们建立了几个脚本文件：每个脚本文件都要用一个独一无二的名字，这样我们才可以从命令窗口执行这个脚本。良好的程序设计习惯要求我们使用具有描述性的名字，例如，计算矩形面积的脚本名为 rect。这种良好的习惯可以帮助我们组织程序文件。此外，关于脚本文件名，MATLAB 还规定了我们必须严格遵循的规则：

- 整个文件名必须是一个单词，不能有空格。如果你想在文件名插入空格，使得文件名更具描述性，则必须使用下划线(shift＋“－”键)。例如，rectange area 不可以作为脚本文件名，因为名字包含一个空格，rectange_area 是合适的脚本文件名。
- MATLAB 中具有特殊意义的字符不允许出现在文件名里。像“＋”、“－”、“/”、“*”和“.”都不可以出现在文件名中，MATLAB 会把它们理解为数学运算符。注意，句点也不可用在文件名中。虽然 MATLAB 给所有的脚本文件自动添加默认的.m 扩展名，但是不允许在文件名中再出现额外的句点符号。
- 文件名的首字符必须是 26 个字母之一。数字及其他“特殊”字符不可以作为文件名的首字符。
- 文件名不能与 MATLAB 中预先定义的函数同名。与 MATLAB 中预先定义的函数名同名会在运行时引起问题。想要知道准备使用的文件名是否与 MATLAB 内部已存在的函数名同名，可以用在线帮助看看这个函数名是否存在。

- 文件名是区分大小写的。如果一个文件名是 Area，则在命令窗口里输入 area，或 AREA 都不会执行这个文件。

选择脚本文件名的最好的建议是，使用简短的、描述性的、独一无二的单词。除了下划线外，不要使用其他任何字符。这些命名规则同样适用于变量名和下一节将要介绍的函数文件名。

3.4　教程：MATLAB 函数文件的使用

在第 3.3 节里，我们讨论了 MATLAB 中一类被称为脚本的程序。本节将讨论 MATLAB 中另一类被称为函数的程序。我们已经使用过 MATLAB 几个内置函数。例 3.3 的 sin 和 cos 函数，表 3-2 的函数以及例 3.5 里使用的 input( )函数都是 MATLAB 预先定义的函数。函数具有以下特征：

- 函数需要输入参数，输入参数放在函数名后的括号里。
- 函数返回一个值或输出值，这个值通常依赖于输入值，并赋给一个变量名。
- 函数中的变量是局部变量，而不是全局变量。与脚本不同，函数不能访问工作区里的全局变量。同样道理，函数中定义的变量不能在工作区中使用。我们称函数中定义的变量为局部变量，因为，工作区中其他脚本和函数不能访问这些局部变量。函数不能通过共享变量与工作区进行通信。它只能通过输入参数和输出值与工作区进行通信。

现在我们通过一个函数例子说明函数与脚本的区别。

例 3.6

一个用户想计算任意一个矩形的面积。我们可以设计一个函数实现此功能。

解：

为了计算矩形面积，这个函数要求已知两个输入参数：矩形的底和高。因此，这个函数的形式是：

```
area_rect(base,height)
```

建立函数与建立脚本使用同一个编辑器。从 MATLAB 运行平台，新建一个 m-file 文件，在编辑器里输入以下函数：

```
1 function A=area_rect(base,height)
2 % This function takes the user-defined base
3 % and height of a rectangle, and computes area
4 A=base*height;
```

这段程序不是一个脚本，而是一个非常简单的函数。因此，我们不能从编辑器的“保存并执行”快捷命令执行这个函数。我们必须先保存这个文件，方法是从选择 File | Save 命令，或者直接单击工具栏上的 Save 图标。有一点非常重要，即函数保存的文件名必须与函数名同名，否则函数就不能起作用。把这个函数保存为 area_rect 文件。当保存文件时，编辑器将使用这个默认的文件名。

虽然这个函数很短也很简单，但是它包含函数的重要成分。我们仔细分析这个函数的第一行内容：

- 函数的第一行的行首必须是 function 单词。它是所有的函数的第一个单词。当 MATLAB 解释程序时，就是根据程序开头的 function 单词区别脚本和函数。
- 位于 function 的右侧、赋值号的左则的变量是函数的输出变量。函数执行后，它是函数返回给工作区的值，函数执行结束后，把这个值赋给输出变量。
- 赋值号右侧的单词是函数名。它同时也是函数保存的文件名。我们就是通过个名字调用函数。
- 函数名后面括号里的变量是输入变量。通过这些变量，我们把数值传递给函数。是函数外部唯一能够传递给函数的值。

总之，在 MATLAB 里，每个函数的第一行必须使用下面的格式：

function *output_variable* = *function_name*(*input_variable_1, input_variable_2, ...*)

除第一行外，函数与脚本没有区别。这个例子的函数包含两行注释内容，只有一行计算语句。如果有必要，一个函数可以包含很多行。命令窗口中所有数学运算符和函数都可以用在函数里。

为了调用函数，我们返回命令窗口。为了清楚说明函数与脚本的区别，我们将从一个“干净的”工作区开始。为此要在命令窗口的提示符后输入 clear 命令。

在提示符后输入以下命令，以最简单的形式调用此函数：

```
>> area_rect(6.5,2.1)
```

这个命令将执行 area_rect( )函数，把 6.5 赋给变量 *base*，把 2.1 赋给变量 *height*。当我们按下 Enter 键时，函数开始执行。由于命令后面没有分号，因此输入变量的值将回显在命令窗口：

```
>> area_rect(6.5,2.1)
ans =
     13.6500
```

切换到工作区窗口，我们看到如图 3-18 所示的内容。

Current Directory | Workspace

Name	Value	Min	Max
ans	13.65	13.65	13.65

图 3-18

需要注意的是，虽然在函数内部我们定义了 *base*、*height* 和 *A* 三个变量，但是它们并没有出现在工作区里。这正好说明了它们是局部变量。这些变量只能在函数内部使用，在函数执行结束后它们就会消失。既然在一个函数内部定义的变量只是在局部起作用，因此我们无需担心它们在函数中与其他变量或函数同名或出现变量名重用问题。

在调用这个函数时，我们并没有把这个函数的输出值赋给某个变量。因此，MATLAB

把它赋给一个默认的临时变量 *ans*。假如我们希望把函数值赋给一个变量，则要用下面的命令：

```
>> Area=area_rect(6.5,2.1);
```

当我们执行这个命令时，MATLAB 把函数的输出值赋给一个全局变量 *Area*，我们在工作区里可以看到这个变量。

调用函数时也可以把变量当作输入参数。为此，我们在命令窗口定义以下两个全局变量：

```
>> B=6.5;
>> H=2.1;
```

现在以这两个变量为输入参数，调用此函数：

```
>> Area1=area_rect(B,H)
```

由于后面没有分号，因此结果输出到屏幕上，并且工作区里新增一个全局变量。

当然，既然在命令窗口输入的任何命令都可以作为脚本的语句，因此可以从脚本调用这个新建的函数。为此，我们新建一个 m-file，输入以下脚本：

```
1 rb=6.5;
2 rh=2.1;
3 ra=area_rect(rb,rh)
```

用“保存并执行”快捷命令，执行这个脚本程序(选择一个文件名，保存此脚本程序)。注意命令窗口和工作区的输出值。

函数的另一个令人感兴趣的特性是，它可以成为 MATLAB 在线帮助菜单的一部分。为了说明这个特性，我们在命令提示符后输入以下命令：

```
>> help area_rect
```

按 Enter 键后，紧跟在函数定义第一行(function 行)之后的注释内容将会出现在帮助文档里。

利用这个特性，可以帮助我们(或其他用户)记住函数的功能和调用方法。

3.5 教程：一维数组的使用

在本章前面几节里执行的运算只是给变量赋单个值。在许多应用程序里，我们希望使用数组，它不同于单个数值。其中的道理等我们读了后面几章后就会明白。目前，我们只学习如何给数组赋值、如何使用数组。

在开始本教程之前，首先用 clear 命令清除工作区。我们要定义一个一维数组，它可以表示很多数值 。为此，在命令行输入以下表达式：

```
>> a=[1 3 6.2]
```

按 Enter 键后，这个赋值运算的结果回显在屏幕上：

```
>> a=[1 3 6.2]
a =
        1.0000 3.0000 6.2000
```

我们注意到，这个命令已把 3 个值赋给变量 a。这可以从工作区得到证实。

输入以下命令，建立第二个数组：

```
>> b=[0.7; -3; 0]
```

同样当我们按 Enter 键时，结果将被回显到屏幕上，如下所示：

```
>> b=[0.7; -3; 0]

b =

     0.7000
    -3.0000
          0
```

注意变量 a 与变量 b 的区别。赋给数组 a 的列表称为行数组。它包含了水平一行的值。数组的各元素可以用空格(本例使用空格)或逗号分隔。赋给数组的是一个列数组。它包含垂直一列的值。注意，在定义列数组时，行与行之间用分号分隔。

当使用数组时，要考虑两个问题：如何给数组赋值、如何使用数组的值。在第 1 章里曾提到这两个问题。当引用变量 a 时，我们是指整个列表值。然而，我们经常需要访问列表中某一个值，或者对某个值进行运算。在这些情形，我们用列表中的索引(元素序号)引用数组中某个元素。MATLAB 的索引从 1 开始。

例如，假如我们想把数组 a 的每个值翻一倍。

要输入以下的命令：

```
>> a=2*a
```

这个命令的作用是，“把变量 a 里的每个值都乘以 2，然后把它赋给变量 a”。列表中每个值都乘上 2，并用新的值覆盖原来的值。这个命令再次强调了“赋值号”与代数中的“等号”的区别。在代数中，像这样的式子“a=2a”是不能成立的(除非 a 碰巧为零)，但是在程序设计中，“a=2*a”是一个有效的表达式，不管 a 的当前值是多少。

按下 Enter 键，就可以看到结果。

```
>> a=2*a
a =
        2.0000 6.0000 12.4000
```

现在假设，我们只想把列表中第二个值恢复到原来的值(3.000)，要输入下面的命令：

```
>> a(2)=.5*a(2)
```

这个命令的作用是“取出数组的第二个元素的值，并把它乘以 0.5，再把结果赋给数

组的第 2 个元素”，a(2)括号里的数值 2 是表示对数组 a 的第二个元素进行运算。

要看到运算结果，需要再次按下 Enter 键：

```
>> a(2)=.5*a(2)
a =
        2.0000 3.0000 12.4000
```

在继续讨论下一个问题之前，我们先仔细分析变量 *a* 后面括号里的数值的作用。它是对数组 a 的某个元素的引用。现在，a 数组的元素值如下所示：

a(1)	*a*(2)	*a*(3)
2.000	3.000	12.400

我们用这种方法访问数组中的某个元素。在这个例子里，a 数组没有 a(4)和 a(5)等元素。另一个需要注意的是，由于括号里的数字表示数组里某个元素的位置，因此像 a(-1)、a(0)和 a(1.34)等都是不正确的。数组的第一个元素都是从 1 开始，第二个元素是 2，依此类推。

现在，我们输入并分析以下命令：

```
>> a(4)= 2*b(1)
```

换句话说，这个命令相当说“把数组 b 的第一个元素乘以 2，并把结果赋给数组 a 的第 4 个元素”。按 Enter 键，注意输出结果：

```
>> a(4)=2*b(1)
a =
       2.0000 3.0000 12.4000 1.4000
```

我们发现数组 a 新增了第 4 个元素，现在它的值如下：

a(1)	*a*(2)	*a*(3)	*a*(4)
2.0000	3.0000	12.4000	1.4000

试试输入下面这个命令：

```
>> b(7)=3
```

按下 Enter 键执行这个命令：

```
>> b(7)=3
b =
    0.7000
   -3.0000
         0
         0
         0
         0
    3.0000
```

我们发现，3.000 0 被添加到数组 b 的后面作为它的第 7 个元素。由于我们还没有定义 b 的第 4 个~第 6 个元素的值，因此 MATLAB 将把默认值 0 赋给它们。

现在我们讨论在运算中使用数组时经常遇到的一些问题。输入下面的命令：

```
>> b(5)=2*a(0)
```

这个命令的含义是“把数组 a 的第 0 个元素值乘以 2，把结果赋给数组 b 的第 5 个元素”。按下 Enter 键，注意它的错误提示信息：

```
>> b(5)=2*a(0)
??? Subscript indices must either be real positive
integers or logicals.
```

错误信息表示“不能执行这个运算”，根据它的提示信息，我们知道，数组名 a 后面括号里的数值必须是大于零的整数，因为它表示元素在数组中的位置。由于根本不存在“零”位置，因此这个命令是毫无意义的。

现在输入以下命令，并按 Enter 键：

```
>> a(1)=3*b(10)
```

这个命令的作用是“取出保存在数组 b 里的第 10 个元素值，乘以 3，再把结果赋给数组 a 的第 1 个元素”，我们又看到错误信息：

```
>> a(1)=3*b(10)
??? Index exceeds matrix dimensions.
```

错误的原因是，我们试图访问数组 b 的第 10 个元素。而实际上，数组 b 只有 7 个元素，这是因为我们试图引用一个并不存在的变量。

本节介绍的错误都是我们在 MATLAB 里使用数组时经常遇到的问题。随着我们的程序越来越复杂，知道并理解这些错误消息可以帮助我们查找程序中存在的逻辑错误。

虽然前面介绍的例子都是在命令提示后使用数组，但是实际上，在脚本和函数里也可以使用数组。我们在下面的例子里介绍数组在函数中的应用。

例3.7

一个用户希望能够计算出任意矩形的面积和周长。要求我们设计实现此功能的一个函数。

解：

为了计算任意矩形的周长，我们本来也可以建立一个与例 3.6 解类似的函数。但是，在本例里，函数的输出变量用一个数组来表示，它的第一个元素表示矩形的面积，第二个元素表示周长。

新建一个 m-file，按下面的形式建立一个函数：

```
1  function AP=area_perim(base,height)
2  % This function computes area and perimeter
3  % for a rectangle defined by base and height.
4  % The output variable takes the form:
5  % [area, perimeter]
6
7  %Compute area
8  AP(1)=base*height;
```

```
9 %Compute perimeter
10 AP(2)=(2*base)+(2*height);
```

保存这个函数(记住，文件必须取 area_perim)。返回命令窗口，输入下面的命令调用这个函数：

```
>> area_perim(6.5,2.1)
```

按下 Enter 键，执行这个命令。我们发现返函数返回一个包含两个值的变量。正如函数所定义的那样，第一个数是矩形的面积，第二个数是矩形的周长：

```
>> area_perim(6.5,2.1)
ans =
     13.6500 17.2000
```

注意，我们也可以用一个数组作为输入参数。打开 area_perim( )函数，把它修改成如下的样子：

```
1  function AP=area_perim(Dimensions)
2  % This function computes area and perimeter
3  % for a rectangle defined by base and height.
4  % The output variable takes the form:
5  % [area, perimeter]
6
7  %Compute area
8  AP(1)=Dimensions(1)*Dimensions(2);
9  %Compute perimeter
10 AP(2)=(2*Dimensions(1))+(2*Dimensions(2));
```

保存修改后的函数。注意在这个函数里，把输入参数重新定义为一个数组。它的第一个元素是矩形的底边长，第二个元素是矩形的高。现在我们要调用这个函数。首先定义一个变量作为函数的输入变量，这个变量的第一个值是底边，第二个值是高。

现在执行这个函数，用 *dims* 变量作为它的输入参数。这一次我们把函数的输出值赋给一个变量，命令如下：

```
>> a_and_p=area_perim(dims)
```

按下 Enter 键，看到如下的输出结果：

```
>> a_and_p=area_perim(dims)
a_and_p =
     13.6500 17.2000
```

3.6 教程：二维数组的使用

在第 3.5 节中，我们介绍了如何用一维数组进行计算。现在我们介绍如何用二维数组进行计算。二维数组，又称为矩阵，由行和列组成。矩阵运算广泛应用于工程计算中，特

别是在有限元、复杂的应力分析、控制系统工程中。第 7 章将详细介绍矩阵运算。本节只简单介绍矩阵在 MATLAB 里的表示。

用 clear 命令清除工作区。首先输入下面的命令定义一个矩阵 **A**：

```
>> A=[1 -1 3; 4 -2 -4; 2 3 1; 0 -3 0]
```

这里的分号表示一行的结束另一行的开始的位置。按下 Enter 键，矩阵定义为如下的形式：

```
>> A=[1 -1 3; 4 -2 -4; 2 3 1; 0 -3 0]
A =
        1  -1    3
        4  -2   -4
        2    3    1
        0  -3    0
```

我们称这个矩阵为(4×3)矩阵，表示它包含 4 行 3 列。注意变量 *A* 在工作区窗口的排列方式。在 value 一列里显示它的大小，如图 3-19 所示。

Current Directory　Workspace

Stack: Base

Name	Value	Min	Max
A	<4x3 double>	-3	4

图 3-19

矩阵中任意一个元素都有一个独一无二的行/列坐标，通过这个坐标可对该元素进行访问。例如，要显示位于第 2 行第 3 列的元素，输入以下命令：

```
>> A(2,3)
```

利用行/列坐标，我们可以引用矩阵的某个元素，或者给某个元素赋一个新值。输入以下命令：

```
>> A(4,1)=-A(2,3)
```

这个命令将读取矩阵 **A** 的第 2 行第 3 列元素，取它的相反数，再把结果赋给矩阵 **A** 的第 4 行第 1 列元素。计算结果如下：

```
>> A(4,1)=-A(2,3)
A =
        1  -1  3
        4  -2 -4
        2   3  1
        4  -3  0
```

用类似的坐标地址，我们还可以对矩阵的整行或整列进行运算。例如，如果要把这个矩阵的第二行全部置零，要用下面的命令：

```
>> A(2,:)=[0 0 0]
```

MATLAB 把这里的 A(2, :)符号理解为“第二行，全部列”。同样道理，我们可以把这个矩阵的第二列和第三列之和赋给第一列：

```
>> A(:,1)=A(:,2)+A(:,3)
```

在这个命令里，MATLAB 把 A(:, 1)符号理解为“第一列的所有行”。

除了在命令窗口输出或修改矩阵外，我们还可以利用 MATLAB 的数组编辑器的功能输出或修改矩阵。用数组编辑器输出某个变量，只需双击它在工作区里名称列中的数组符号，如图 3-20 所示。

Name	Value	Min	Max
A	<4x3 double>	-3	4

图 3-20

这会打开数组编辑器，显示该数组的值，如图 3-21 所示。

	1	2	3
1	1	-1	3
2	4	-2	4
3	2	3	1
4	0	-3	0

图 3-21

用数组编辑器不仅可以显示数组的内容，还可以在这个窗口修改数组的值。单击位于第 3 行、第 3 列的元素，把原来的值 1 改为 10，如图 3-22 所示。

	1	2	3
1	1	-1	3
2	4	-2	4
3	2	3	10
4	0	-3	0

图 3-22

按下 Enter 键，确定修改。修改后的变量包含了这个新值。要关闭数组编辑器，只要单击右上角的“×”符号，如图 3-23 所示。

图 3-23

为了验证矩阵 **A** 的内容是否已发生变化，可在命令窗口输入以下命令，显示它的值：

```
>> A
```

在许多实际应用中，矩阵变量在脚本和函数中非常有用。在下面这个例子里，我们将演示矩阵变量在脚本中的使用。

例 3.8

建立一个脚本程序，计算表 3-4 中(重复例 3.4 中的数据)所有矩形的面积和周长。

表 3-4　例 3.4 矩形尺寸

	底边	高
矩形 1	6.5in	2.1in
矩形 2	7.2in	3.0in
矩形 3	7.5in	3.3in

解：

作为演示，我们建立了一个矩阵变量，包含了表 3-4 的全部信息，然后使用例 3.7 建立的 area_perim( )函数求矩形的面积和周长。

新建一个 m-file，输入以下的脚本程序：

```
1  %Define the base/height array
2  Dims=[6.5 2.1; 7.2 3; 7.5 3.3];
3  % Compute area and perimeter of Rectangle 1,
4  % and store them in the first row of Props
5  Props(1,:)=area_perim(Dims(1,:));
6  %Repeat for Rectangle 2, and store in second row
7  Props(2,:)=area_perim(Dims(2,:));
8  % Repeat for Rectangle 3
9  Props(3,:)=area_perim(Dims(3,:));
10 %Output results
11 Props
```

保存并执行这个脚本文件，给它一个合适的文件名。注意，第 11 行行尾没有分号，而且只有一个矩阵变量名，它存放了全部矩形的面积和周长。用这种简单方法显示变量的值：

```
Props =
    13.6500   17.2000
    21.6000   20.4000
    24.7500   21.6000
```

变量 *Props* 的第一列是矩形的面积，第二列是矩形的周长。

3.7 教程：保存 MATLAB 会话过程

在用 MATLAB 进行计算时，在会话过程中命令窗口或脚本文件中定义的全部变量都可以保存起来供以后使用，除非：

- 执行了 clear 命令，或者
- MATLAB 会话过程由程序关闭

有时候，我们希望保存部分或全部变量，供以后的 MATLAB 会话过程使用。本节将介绍 MATLAB 如何用 save 和 load 命令实现数据保存功能。

假设读者已经执行了第 3.6 节的全部命令，那么就已经在工作区里定义了 4 个变量，如图 3-24 所示。在这里变量 *ans* 对我们没有用处，当需要把脚本文件的计算结果输出到屏幕时，MATLAB 自动创建这个默认变量，它复制了变量 *Props* 的内容。前面我们曾用 clear 命令清除整个工作区， 也可以用 clear 命令清除一个或多个不需要的变量。只清除工作区的 *ans* 变量，在命令窗口输入以下的命令：

```
>> clear ans
```

在 clear 命令后跟一个变量名，就可以有选择地清除某个变量。注意，现在工作区里不再有 *ans* 变量。

剩下的 3 个变量可以在未来的 MATLAB 会话中使用。使用 save 命令，可以使我们用 MATLAB 专用的格式保存这 3 个变量。必须选择一个合适的名字保存这些变量。在这个例子里，输入以下的命令，把这 3 个变量保存到一个名为 rectangles.mat 的文件里：

```
>> save rectangles
```

现在工作区中全部变量都保存到工作目录里的 rectangles.mat 文件里。MATLAB 自动添加默认的扩展名.mat，因此它不需要在命令中输入。

用 load 命令可以随时恢复这些变量。为了说明 load 命令的用法，我们先用 clear 命令清除工作区：

```
>> clear
```

现在我们发现工作区是空的。把保存的变量恢复到工作区，输入 load 命令，如下所示：

```
>> load rectangles
```

现在 *A*、*Dims* 和 *Props* 3 个变量已经恢复到工作区。

也可以有选择地保存部分变量。例如，我们只想把 *Dims* 和 *Props* 两个变量保存到名为

rectdata 的文件里，输入以下命令：

```
>> save rectdata Dims Props
```

用 clear 命令清除工作区：

```
>>clear
```

重新装入保存的变量，要用 load 命令(后面跟合适的文件名)：

```
>> load rectdata
```

现在 *Dims* 和 *Props* 两个变量已经被恢复到工作区里。

3.8　习题

1. 用 MATLAB 计算图形(见图 3-24~图 3-26)的面积和周长。

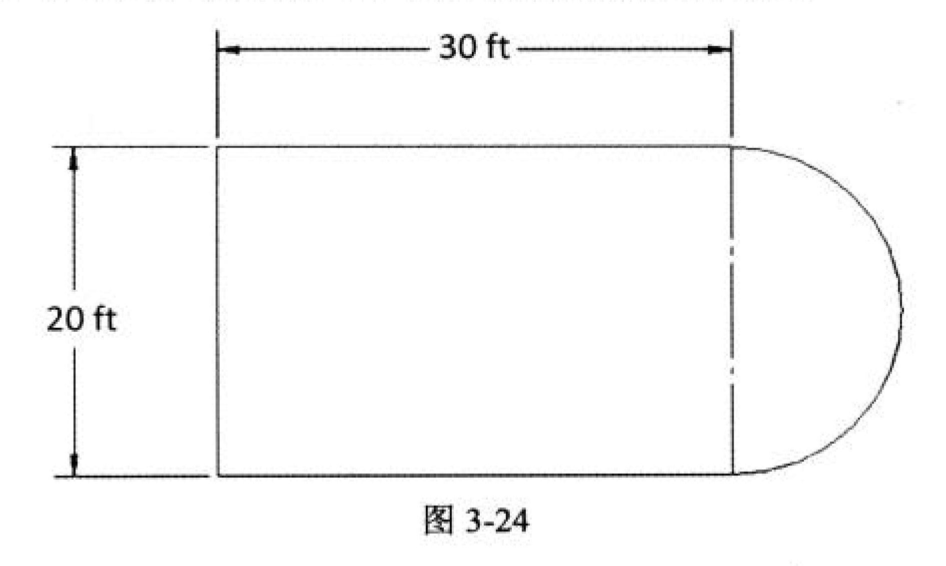


图 3-24

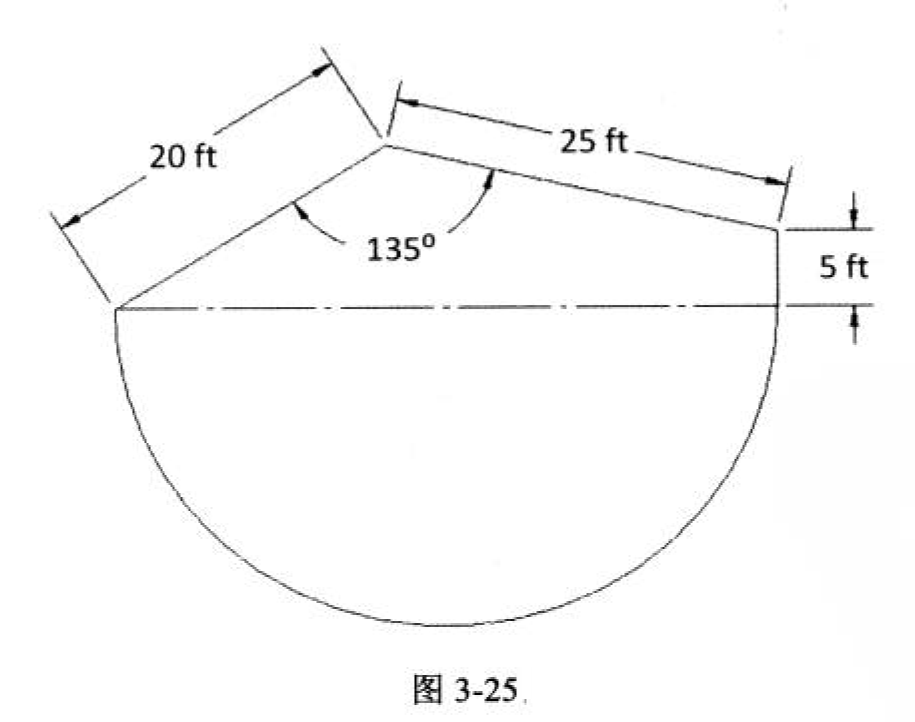


图 3-25

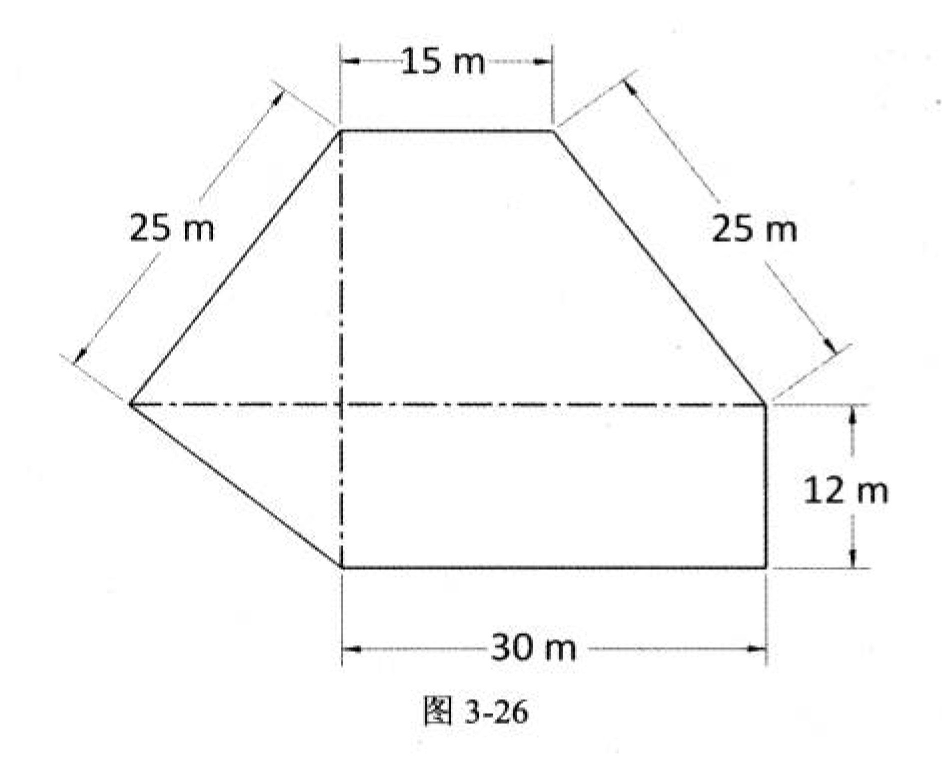


图 3-26

2. 图 3-27 是一个圆锥形储油罐。它存满了水。用 MATLAB 计算：
   (a) 罐里水的体积。
   (b) 储满水时的重量。

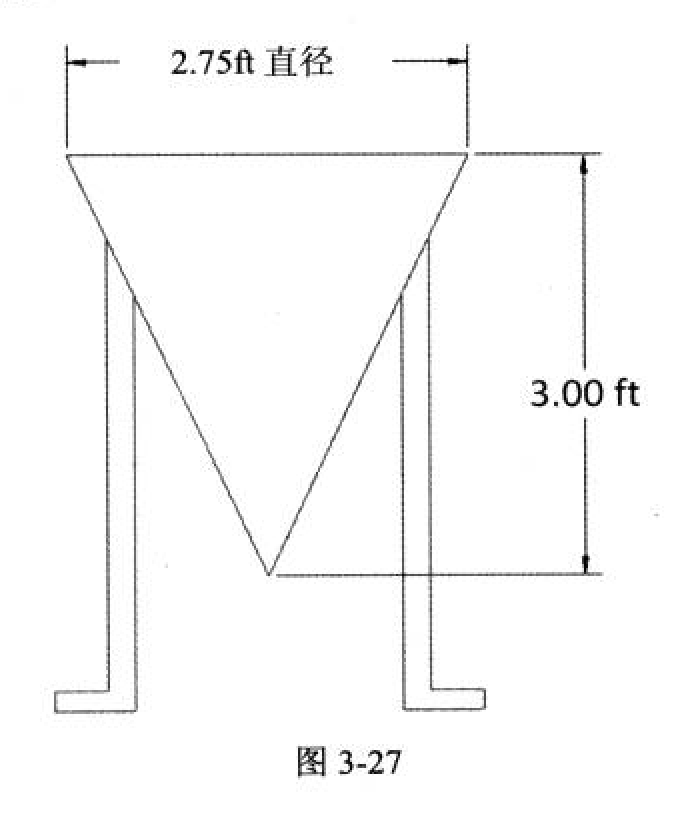


图 3-27

3. 用 MATLAB 计算，容积等同于习题 2 的圆锥储油罐的球体的半径。

4. 习题 2 中的油罐制造商打算设计另一个模型，它的容量是图 3-27 所示容器容量的两倍。高度与直径的比例不变，用 MATLAB 确定新模型的直径和高度。

5. 习题 2 中的油罐制造商打算设计另一个模型，它的容量是图 3-27 所示容器容量的一半。高度与直径的比例不变，用 MATLAB 确定新模型的直径和高度。

6. 编写一个 MATLAB 脚本，它自动计算任何圆锥形储油罐(见图 3-27)的体积和所储水的重量。用 input( )函数输入直径和高度。

7. 编写一个 MATLAB 脚本，它提示用户输入一个直角三角形的两条直角边的长度(即输入图 3-7 的 x 和 y)，用勾股定理计算它的斜边长。

8. 编写一个 MATLAB 脚本，它提示用户输入一个直角三角形的斜边和一条直角边的长度，计算另一条直角边的长度和它的面积。

9. 编写一个 MATLAB 脚本，它提示用户输入一个直角三角形的两条直角边的长度(即输入图 3-7 的 x 和 y)，计算直角三角形的两个内角。

10. 考虑图 3-24，编写一个 MATLAB 脚本，它提示用户输入矩形部分的底边和高。然后根据输入值，计算整个图形的面积和周长。

11. 考虑第 1.1.1 节里的炮弹发射问题的解析解。编写一个脚本，要求用户输入发射速度和角度，用解析方程计算最高发射高度、飞行时间和水平飞行距离。

12. 编写一个 MATLAB 函数，名为 rad2deg( )，输入为弧度，把它转换为度，并返回。

13. 编写一个 MATLAB 函数，它可以计算任意圆锥形储油罐的体积和所储水的重量(见图 3-27)。函数要有两个输入参数(直径和高度)，返回一个数组，第一个元素是油罐的容积，第二个元素是所储水的重量。

14. 编写一个 MATLAB 函数，执行以下单位换算，要求输入参数是以 SI 为单位的值，把它转换为美国自定义单位，并作为函数值返回。

(a) 把 cm 转换为 in。

(b) 把摄氏度转换为华氏度。

(c) N 转换为 bl。

(d) 把 m/s 转换为 mile/hr。

15. 编写一个 MATLAB 函数，输入参数是一个含有 5 个元素的一维数组，返回值是数组的平均值。

16. 编写一个 MATLAB 函数，取名为 circles( )，它可计算 5 个圆的周长和面积。要求函数有 5 个输入参数 (直径)，返回一个包含 5 个圆的周长和面积的二维数组。以直径 1cm、2cm、3cm、4cm 和 5cm 为例，调用这个函数。

17. 编写一个 MATLAB 函数，取名为 squares( )，它计算 5 个正方形的周长和面积。要求函数有 5 个输入参数 (边长)，返回一个包含 5 个正方形的周长和面积的二维数组。以边长 1cm、2cm、3cm、4cm 和 5cm 为例，调用这个函数。

18. 编写一个 MATLAB 函数，取函数名为 spheres( )。计算 5 个球体的表面积和体积。要求函数有 5 个输入参数 (直径)，返回一个包含 5 个球体的表面积和体积的二维数组。以直径 1cm、2cm、3cm、4cm 和 5cm 为参数，调用这个函数。

19. 编写一个 MATLAB 函数，取函数名为 cubes( )。计算 5 个立方体的表面积和体积。要求函数有 5 个输入参数(边长)，返回一个包含 5 个立方体的表面积和体积的二维数组。以边长 1cm、2cm、3cm、4cm 和 5cm 为例，调用这个函数。

20. 在制造轴承的过程中，轴承的零件(如滚球)需要经过一道硬化工序。具体过程是先加热再迅速冷却，或者把它浸入到油槽或水槽里，这个过程即为淬火。滚球轴承的温度是时间的函数。$T(t)$可以按以下方程估算：

$$T(t)=(T_i-T_\infty)e^{-t/\tau}+T_\infty$$

式中 t 是轴承浸入槽中的时间，单位为秒。T_i 是轴承的初始温度，T_∞ 是油的温度。τ 是

时间常数，单位是 s，它与轴承的材料、轴承的几何形状及油的特性有关。编写一个 MATLAB 函数，以 T_i、T_∞ 和 τ 和 3 个不同时刻的 t 为输入变量，计算并返回在 3 个不同时刻的轴承温度，用一个一维数组表示。

假设时间常数 $\tau = 60$s，假设轴承的初始温度为 1000°C，油的温度为 60°C。求轴承在 1s、10s 和 100s 时的温度。

21. 编写一个 MATLAB 函数，要求：

- 它的输入参数是一个 3×3 矩阵。
- 建立一个新的矩阵，新矩阵的行是输入矩阵的列，列是输入矩阵的行。
- 返回这个新矩阵。

例如，如果输入矩阵为：

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}$$

输出矩阵则是：

$$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}$$

22. 编写一个 MATLAB 函数，它的输入参数是一个 4×4 的矩阵，返回一个 1×4 的数组，数组的每个元素是矩阵的每一行的平均值。

23. 编写一个 MATLAB 函数，它的输入参数是一个 4×4 的矩阵，返回一个 4×1 的数组，数组的每个元素是矩阵的每一列的平均值。



第 4 章

MATLAB 编程

引言

在第 3 章中，我们已经学习了如何用 MATLAB 执行许多计算操作。在这个方面，MATLAB 的使用与计数器的使用、电子表格的使用没有差别。程序设计语言的强大功能在于它可以控制计算过程。本章要学习 MATLAB 程序设计的两个语句：循环语句和选择语句。前者可以使一个程序反复执行计算，后者根据程序变量的判断条件有选择性地执行部分代码。

本章，我们要学习以下内容 ：

- 如何用流程图说明程序的执行步骤。
- 如何用 for 和 while 循环控制 MATLAB 里的程序流程。
- 如何在 MATLAB 程序里添加逻辑分支。
- 如何格化式 MATLAB 输出结果。

4.1　流程图

流程图是用来说明一个程序执行步骤的图形。以前把设计流程图看成是编写程序的一个非常重要的步骤。由于 CPU 的计算时间非常宝贵，在第一次实际运行程序之前，我们先要花大量的时间准备好程序。自从出现了 PC 后，实时调试计算机程序成为了一个标准的约定，因为现在计算机的 CPU 时间几乎总是空闲的。虽然为本书的那些小程序绘制流程图可能显得没有必要，但那是我们掌握流程图的绘制所必不可少的：

- 它们可以帮助我们设计复杂的程序。
- 与词语相比，用流程图更容易说明程序的功能。
- 它广泛应用于其他方面，如用来说明设计和制造过程中的顺序。

图 4-1 是一些标准的流程图符号。虽然美国国家标准委员会定义了一套标准符号，但

是这些符号的用法相差很大。为简单起见，我们只使用3个符号：用矩形框表示程序的执行步骤，用菱形框表示逻辑判断点，用圆表示程序执行路径的连接符。用箭头连接各个步骤，并表示程序行进的方向，即步骤的“流程”。

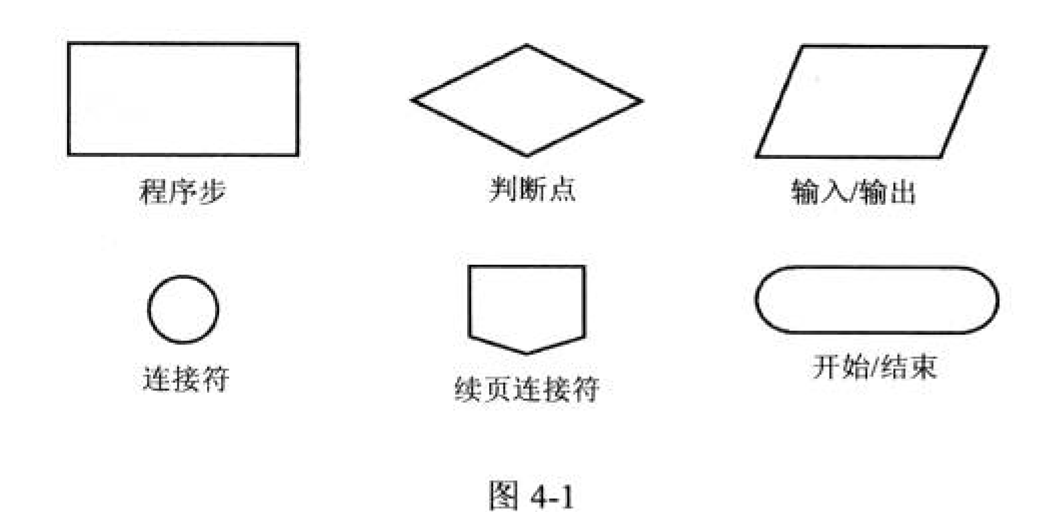


图 4-1

正如前面曾提到，流程图各个符号的用法变化很大。但是，用菱形符号表示决策点是大家都接受的做法。把决策点看成是道路上的交叉口，在交叉口有多个道路通向其他地方。在流程图里，一个决策点有一个箭头流入，多个箭头(通常是两个箭头)流出，如图 4-2 所示。

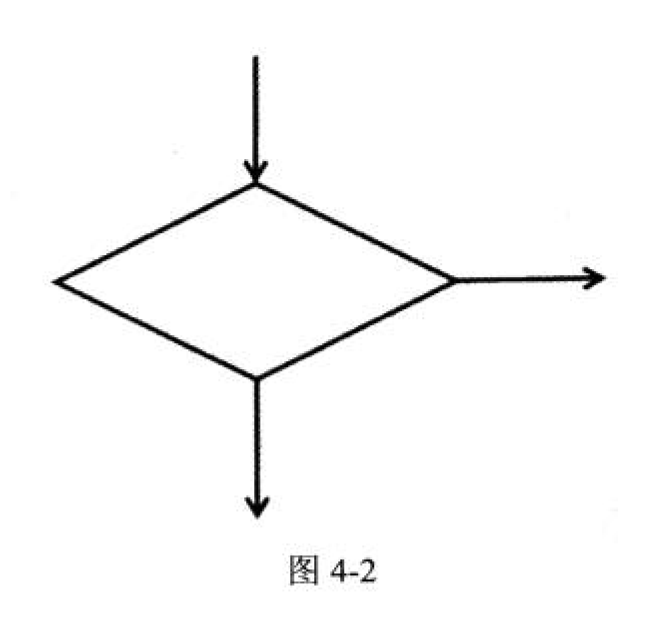


图 4-2

流程图的应用不只限于程序设计。例如，某个公司的新产品设计流程可以用图 4-3 表示。当一个新产品的需求被确定后，工程师用文档记录设计要求(新产品的属性)和约束条件(成本、必须遵循的安全标准、公司的生产能力等)。最初的设计工作需要确定产品的可行性，它是否满足全部要求和约束条件。在这个阶段的最后，需要举行一个设计评审的会议。在评审会议上，可做出决策。如果初步的设计方案可行，则开始进行详细设计，如果不可行，则需要做另一个决策。另外一种可能是完全停止该产品的研发工作。如果项目开始详细设计，则在设计方案确定之后，还需要举行第二次评审。在这次评审会议上，工程和管理人员需要决定该设计方案是否满足所有设计要求和约束条件。现在又到了一个决策点：公司是应该立即投入生产，还是在某些方面进行改进。虽然图 4-3 的流程图是设计过程的一个非常简单的例子，但是它说明了比较复杂的设计过程的主要特征。由于设计是一个循环过程，因此设计过程中经常需要跳回到(loop back)前面的位置重复执行前面的步骤。

此外，我们必须注意到，处理过程在决策点处的流向取决于决策点的逻辑判断。

定义求解要求和约束条件
构思初步解决方案
是一个可行解吗？
否
还继续吗？
是
否
结束求解
是
得到详细求解方案
满足求解要求？
否
是
执行此方案

图 4-3

在设计过程的流程图中，决策点位置的路径选择是由管理人员和工程师们根据提交在设计评估会的设计数据的评估结果做出的。在程序设计中，决策点的路径选择是由逻辑判断的结果决定的。例如，在程序的某个决策点，需要对某个数的绝对值进行判断(我们暂时不要考虑，MATLAB 实际上已内置了一个具有绝对值判断功能的函数)。如果这个数大于或等于 0，则我们不执行任何命令，如果小于 0，则需要改变它的符号。

图 4-4 的决策点符号把一个逻辑判断表示为一个问题。离开菱形的箭头上用 Yes 和 No 表示这个问题的两种可能答案。如果答案是“是”，(即 a 小于 0)，则程序的流程转向下面的一个步骤，这个操作步骤将改变 a 的符号。如果答案是“否”，则跳过这一步骤。不管哪种情形，程序流程都会继续执行到下一个步骤 。注意，这里使用连接器显示多个路径。虽然这不是必需的，但是画出连接器是 MATLAB 程序设计的一个良好的习惯，因为每个连接器对应于 MATLAB 程序的一个 end 语句。

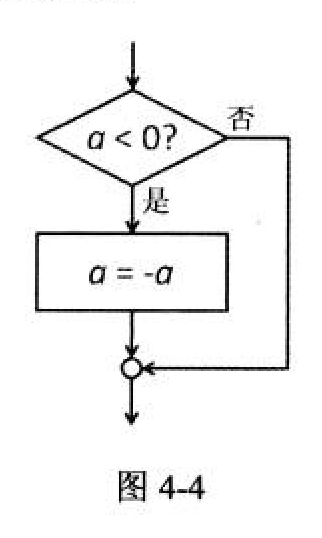


图 4-4

这个流程图中的逻辑判断在 MATLAB 里用 if 语句来实现，在本章的后面将介绍它的用法。我们在第 2 章介绍 Excel 时，曾使用过同样功能的 if 语句。与第 1 章介绍的伪代码一样，流程图中的决策判断是通用形式。正是由于这个原因，我们用流程图保存和说明程序的功能。这样不懂某个特定程序设计语言的人也可以理解这样的流程。

4.2 教程：循环命令

在第 3 章中，为了执行一系列的运算，我们曾编写过一个简单的 MATLAB 程序。然而，我们不得不为每个计算分别编写代码，脚本文件中的每一行对应计算中的每一个步骤，脚本中的步骤顺序执行。重复性非常强的计算需要在脚本中重复输入命令。

在许多情况下，如果只用描述一次多次重复的计算，即只用单独一组指令，不仅比较符合逻辑，而且代码也比较简短。我们只要在程序中表示出重复性计算的重复次数，让计算机执行重复操作。本节我们将介绍两种功能强大的程序结构，这两种结构可以用于重复性的计算：for 循环和 while 循环。for 循环用于重复次数确定的计算，而 while 循环用来执行重复计算，直到某个条件成立为止。在下面几节里，我们将讨论这两类循环的使用方法。

4.2.1 for 循环

在所有编程语言里，for 循环可能是最广泛使用的用来执行重复计算的程序结构。在有些语言里，这个类循环也叫做 do 循环。如果一组计算需要重复执行 10 次，则相应的 for 循环的逻辑结构将如图 4-5 所示。用一个计数器 *m* 记录计算的次数。当计数器的值达到限定值时，程序就从循环语句前进到它后面一组指令。

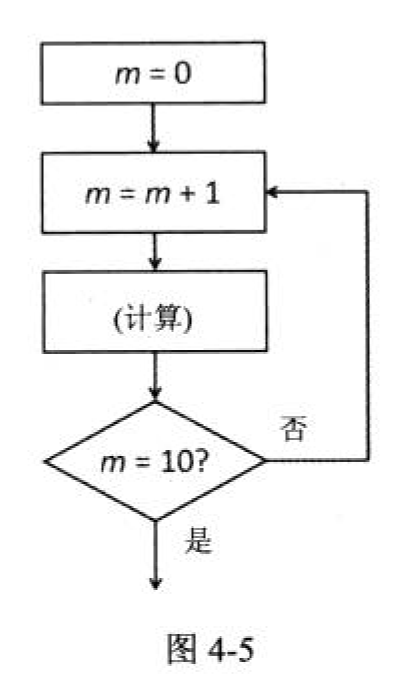


图 4-5

在 for 循环里，计数器的增量和决策点的判断都嵌在同一个命令里。一个简单的 for 命令如下所示：

for counter=1:limit

(Repeat Instructions)

end

实际上，计数器的起始值并不一定是 1，可以取其他值，它的增量也可以是其他值，但是这里我们只是为了使结构简单，使得计数器的值正好对应循环的重复次数。for 循环的流程图可以简化为如图 4-6 所示的形式。

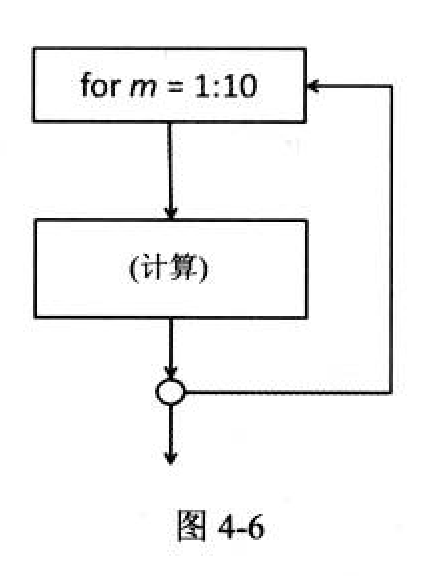


图 4-6

为了说明 for 循环的典型用法，我们启动 MATLAB，然后在脚本文件里输入以下内容：

```
1 a = 0;
2 for m = 1:10;
3  m
4  a=a+m
5  pause
6 end;
```

注意 MATLAB 编辑器会自动缩进 for 命令与 end 语句之间的语句。按这种方式缩进有助于理解和调试脚本程序。另外还要注意的是，第 2 行和第 6 行后面的分号不是必须的。即使这两行后面没有分号，也不会有任何内容输出到屏幕上。

保存这个脚本程序，取文件名为 fortest。

当我们在命令窗口里执行这个脚本时，MATLAB 这样来解释脚本中的语句：

(1) 建立变量 *a*，并且把它置 0。

(2) for 循环开始，变量 *m* 被置 1(即赋值号后的第一个数)。

(3) 执行 for 命令下一行的语句，在本例中：

a. 把变量 *m* 的值输出到屏幕上。

b. 把 *m* 的值与变量 *a* 的值相加，把和赋给 *a*。

c. 程序暂停，用户按任意键继续。(我们之所以增加这一行，是为监视变量的值。实际上，我们不经常使用 pause 命令。)

(4) 当遇到 end 语句时，把变量 *m* 的值与 for 命令里的 limit 值进行比较：

- 如果 *m* 与增量(这里是 1)之和大于 *m* 的上限值 limit(即本例中冒号之后的 10)，就输出循环结果。程序继续执行 end 之后的语句。

- 如果 m 与增量之和没有大于 m 的上限值，就把 m 的值加上增量(本例是 1)，然后重复 for 与 end 之间的语句(步骤 3)。

执行这个脚本程序，跟踪程序执行过程中的每个变量值。当程序执行 pause 命令时，需要按下 Enter 键，程序才会继续执行。当执行结束时，将出现命令提示符(>>)。

在这个例子里，脚本中的第 3 行~第 5 行之间的语句被重复执行 10 次。

当我们既要自动执行重复语句，又需要将计数变量的值作为计算的一个有用部分时，for 循环的真正强大功能才会表现出来。为此，我们举这样一个例子：我们要绘制正弦函数在 0~2π之间的曲线，用 for 循环自动重复执行其中的计算。这个程序的流程图可以用图 4-7 来表示。

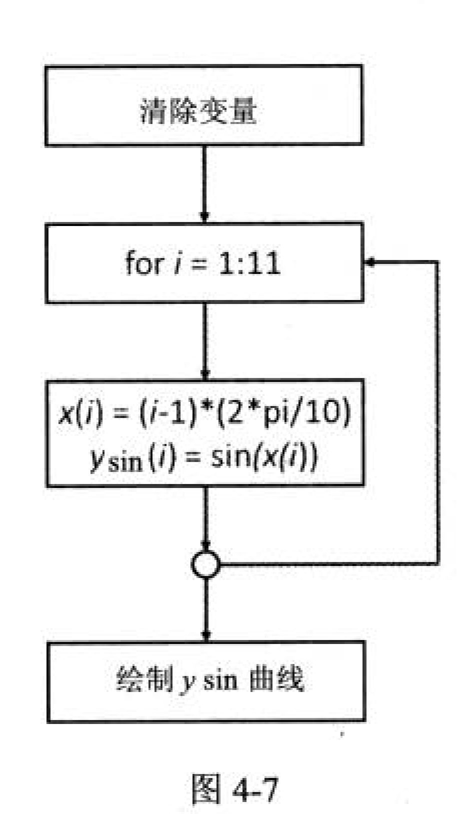


图 4-7

注意这个例子里的变量 i，它既是循环计数器，又是数组的索引变量。利用计数器可以访问一个大数组，每次一个元素，而实际上只需要编写一次代码。我们把 0~2π之间的角度分为 10 等分(赋给变量 x)，需要计算 11 个数据点，它们对应如下的 x 值：

$$x = 0, 0.2\pi, 0.4\pi, 0.6\pi, \cdots 2.0\pi$$

在应用程序中，特别是结果需要绘制图形时，经常遇到这样的情形：自变量 x 从 0，经过 n 个间隔(本例是 10)。在这些程序中，计算数应该从 1 开始开始，逐一递增到 n+1(即 for i=1:n+1)，自变量 x 在每一步的值可以表示为：

$x(i)=(i-1)$ *increment

其中：

increment =(UpperLimit) /n

在本例里，上限(UpperLimit)的值是 2π。

新建一个 m-file，输入以下的脚本：

```
1    % Clear out the workspace
2    clear;
```

```
3
4      % Generate the data sets
5      for i = 1:11;
6           x(i) = (i-1)*(2*pi/10);
7           ysin(i) = sin(x(i));
8      end;
9
10   plot(x,ysin)
```

把文件保存为 sinewave，在命令窗口里输入这个文件名，执行这个脚本。在这个例子里，我们引入了 plot 命令，它在一个新窗口里建立一个图形，这个窗口的标题为 Figure 1，如图 4-8 所示。plot 命令根据两个数组建立图形。第一个数组定义水平轴上的点，第二个数组包含了需要绘制的函数对于水平轴的值。注意，这两个数组必须是一维的，而且必须是同样大小的。在第 5 章中，我们将进一步学习图表的建立、编辑和格式设置。另一个需要注意的是第二行的 clear 命令。我们从第 3 章已经知道，clear 命令可用来清除内存中的变量值。当把一个新值赋给一个标量时，新值会覆盖原来的值。但是当我们把一个值赋给数组中的一个元素时，数组的其他元素不受影响。因此，使用数组时，最好在进行运算之前清除数组变量。

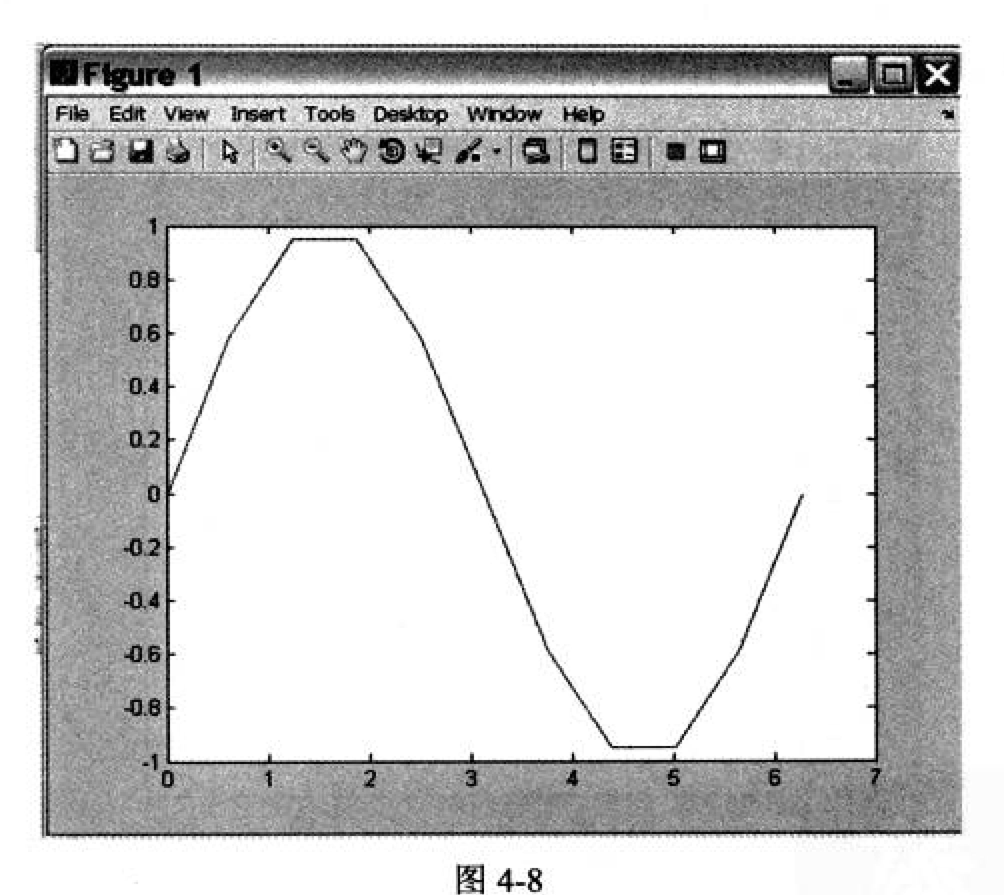


图 4-8

我们得到如图 4-8 所示的图形，但是发现图形曲线并不十分光滑，这是因为 MATLAB 建立的曲线是由许多两个数据点之间的直线段构成的。然而，我们很容易把每个数组的元素个数从 11 变为 1001(1000 步)。打开脚本，把它修改为如下所示的形式(注意第 6 行和第 7 行的变化)：

```
1    % Clear out the workspace
```

```
2     clear;
3     hold off;
4
5     % Generate the data sets
6     for i = 1:1001;
7        x(i) = (i-1)*(2*pi/1000);
8        ysin(i) = sin(x(i));
9     end;
10
11    plot(x,ysin)
```

保存修改后的脚本。

我们可以使用命令窗口的一个快捷键。正如从历史命令窗口可以看到，MATLAB 会记住最近的命令。利用向上移动键向上滚动，可以重复执行历史命令中的任意一个命令，最近输入的命令位于最下面。注意，`如果我们在编辑器里用“保存并执行”命令执行一个脚本，这个命令不会保存在命令历史列表里，则这种快捷方式也不会起作用。

在命令提示符位置，按向下移动键一次，显示脚本文件名 sinewave，按下 Enter 键，执行此脚本。

您是否发现，当数据点更多时，我们得到的曲线也更加光滑、更加准确了(见图 4-9)? 但是程序的复杂性或代码的长度并未增加。

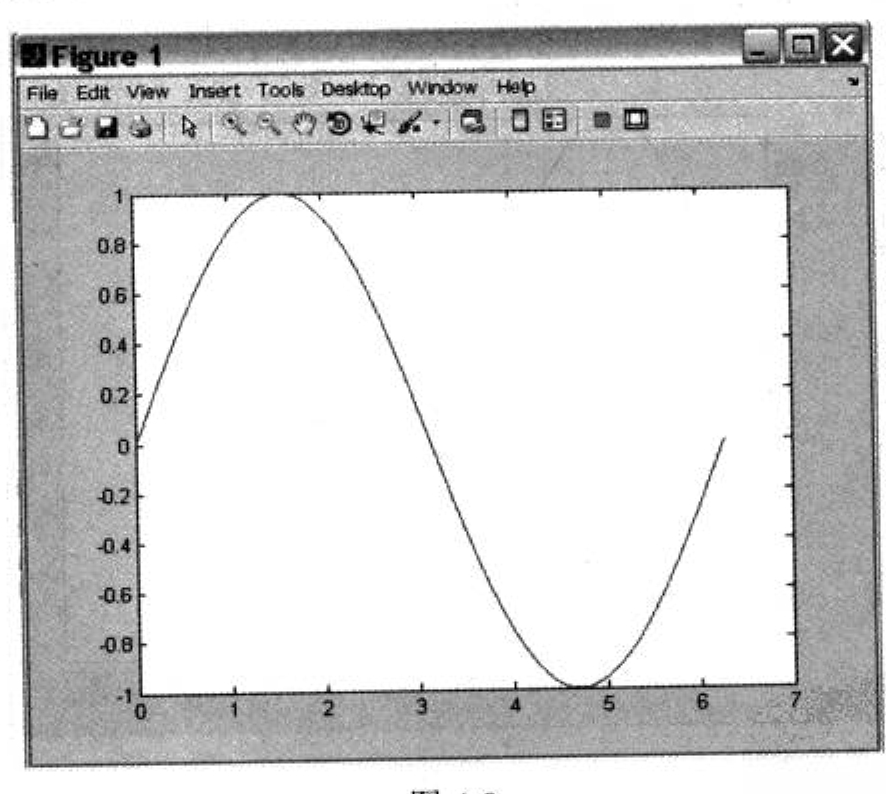


图 4-9

在 for 和 end 语句之间的语句个数或复杂性并没有限制，但是不论什么语句，都是循环的一部分。当我们事先知道重复计算的次数时，for 循环是实现重复性运算的一个强有力的工具。然而，在某些情况下，我们希望反复循环，直到满足某个条件为止。事先并不知道需要重复计算多少次。在这种情况下，我们可以使用 while 循环。

4.2.2 while 循环

while 循环可重复执行一组语句，直到某个变量“击中”某个目标值。while 循环的逻辑关系可以用图 4-10 来说明。与 for 循环不同，while 循环不需要计算数变量。但是在很多情况下，我们总是希望添加一个计数器，因为循环中程序执行的次数通常对我们十分重要。while 循环包含一个逻辑条件，它控制循环的次数。只要这个条件成立，循环就继续进去下去。使用 while 循环需要考虑的一个因素是可能会建立一个无限循环。很可能，我们无意中定义一个永远不会成为 false 的条件。如果发生这种情况，程序会一直运行，直到强制终止程序运行为止。按下 Ctrl+C 组合键可以终止 MATLAB 程序的运行。在 PC 上，这很不方便。对于在主机上运行的复杂程序，这种错误的代价可能会很大。

MATLAB 的 while 循环命令在第一行里包含一个逻辑条件：

```
while(condition)
    (Repeat instructions)
end
```

while 循环的流程图可以简化为如图 4-11 所示的形式。

作为一个例子，我们考虑如图 4-12 所示的算法。

建立一个 m-file 文件，输入以下脚本程序：

```
1 % Initialize variables
2 m = 0;
3 a = 0;
4
5 while a < 54;
6       m = m + 1
7       a = a + m
8       pause
9 end;
10
11 a
```

把这个脚本保存为 whiletest 文件，并运行这个程序。当我们执行这个脚本时，需要注意到，它的功能与 fortest 脚本完全一样。不同之处是在循环结束位置，会监视变量 *a* 的值。还需注意的是，在 while 循环里，变量不会自动递增。如果我们希望跟踪循环执行次数，必须定义一个变量作为计数器(*m*)，在循环中，让这个变量增加 1(第 6 行)。在这种情况下，*m* 的最终值是 10，它表示循环中的程序执行了 10 次。

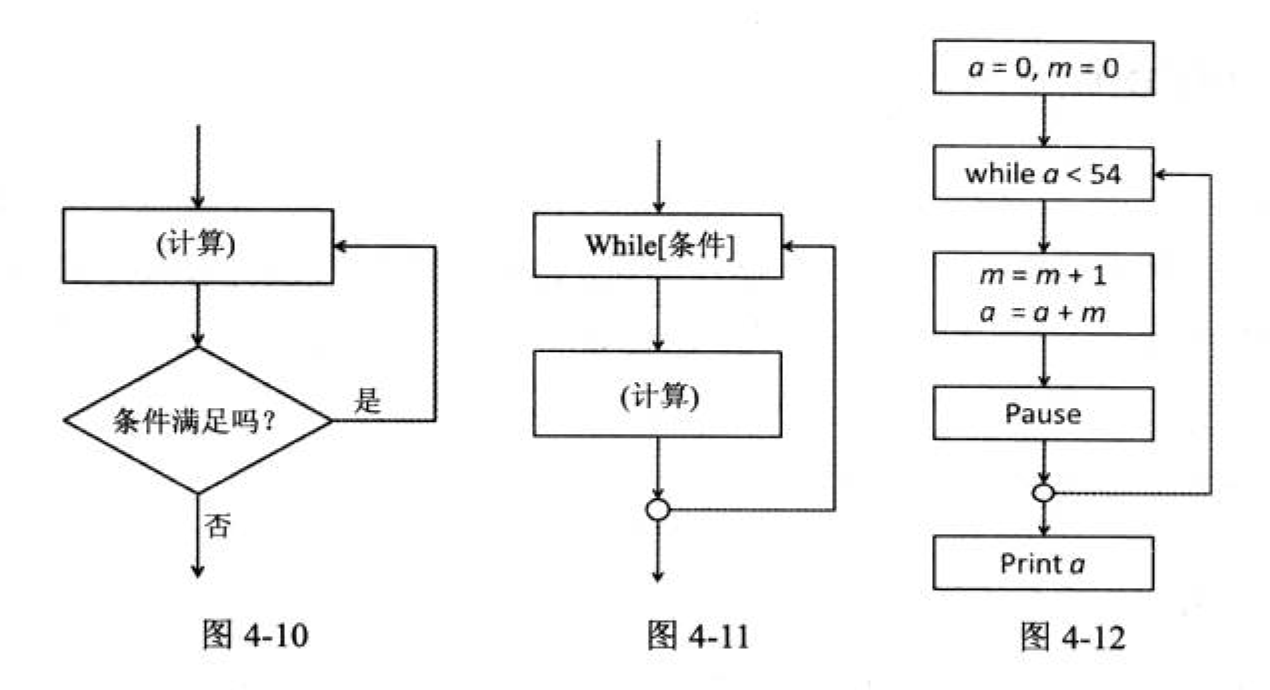


图 4-10　　图 4-11　　图 4-12

注意，如果我们修改“终止条件”，则循环执行的次数也会跟着改变。例如，把这个脚本修改成如下的形式(注意在第一行再增加一个 clear 命令，并修改 while 一行的内容)。执行这个脚本程序。

```
1 clear;
2 % Initialize variables
3 m = 0;
4 a = 0;
5
6 while a < 100;
7        m = m + 1
8        a = a + m
9        pause
10 end;
11
12 a
```

注意，在执行了第 14 次循环之后，a 的值变为 105。因为 while 行的逻辑条件($a<100$)不再成立，因此，程序结束。

在 while 行，我们在小于(<)条件之外，可以再增加其他条件。例如，把这个脚本修改为如下的内容：

```
1 clear;
2 % Initialize variables
3 m = 0;
4 a = 0;
5 b = 200;
6
7 while a < 100 & b >= 50;
8        m = m + 1
9        a = a + m
10       b = b - 10
11       pause
12 end;
```

```
13
14 a
15 b
```

保存并执行这个脚本，监视输出结果。只要这两个条件都成立(即 *a* 小于 100，*b* 大于或等于 50)，这个循环就一直执行。作为一个练习，把 *a* 和 *b* 的初始值设置为其他值，仔细观察 *a* 和 *b* 初始值的变化对循环执行次数的影响。

4.3 逻辑判断语句

本节我们继续讨论 MATLAB 编程的另一个基本结构，即逻辑分支语句。利用这种结构，我们根据某个程序变量的逻辑条件，可以有选择地执行部分代码。所有的逻辑语句都可以认为是传统的 if-elseif-else 逻辑结构。在下面的小节里，我们要分析这种结构。

4.3.1 if 语句

if 语句是其他有逻辑分支语句的基础。它对应于我们平常使用的的 if-then 逻辑语句："如果一个变量取某个值，则程序就执行某几行代码。"

在 MATLAB 里 if 语句的结构如下：

```
if(条件)
  (只有条件成立时才执行的几个语句)
end
```

为了说明 MATLAB 的 if 语句的使用，应启动一个新 m-file 文件，输入以下脚本：

```
1     % Get the user input
2     y = input('Enter a number less than or equal to 10: ');
3
4
5      % If the user entered a number bigger than 10,change the value to 10
6      if y > 10
7           fprintf('The number you entered is greater than 10. It will be
       changed to 10\n')
8           y = 10;
9     end;
10
11    % Display the number
12    y
```

在这段脚本里，我们引入了一个新的 MATLAB 命令——frpintf，把结果按正确的格式化输出到屏幕是有很必要的。fprintf 命令的最后一个字符(\n)表示跳转到下一行。在本章的稍后还要更详细地讨论这个命令的用法。在这段脚本里，我们还要注意第 2 行、第 5 行和第 7 行。由于页面的空间的限制，这些行都占用了两行。但当我们在编辑器中输入这些行时，它们都只是占用一行。

程序中的逻辑判断可以用图 4-13 来说明。如果 if 语句里的判断条件(在本例，即当 $y>10$)成立，则执行位于 if 与 end 之间的全部语句。如果条件不成立(在本例，即当 $y\leqslant10$)，则程序跳过 if 与 end 之间这段代码，执行 end 之后的语句。我们之所以称这样的结构为条件分支是因为程序根据逻辑判断的值，“分支”到两段不同的代码。

保存这个脚本，取名为 iftest，多次运行这个脚本文件，验证这个选择结构的正确性。

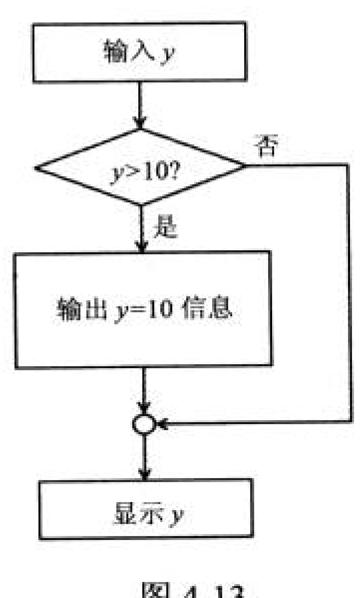


图 4-13

当然，if 语句中的逻辑条件可能会比本例复杂许多。我们把经常使用的关系运算符列在表 4-1 中。利用逻辑运算符，可以把关系运算组合起来，这些逻辑运算符也列在同一个表里。例如，考虑下面的语句：

```
if x==5
```

则只有当 x 等于 5，才会执行 if 语句后面的命令。注意，单个等于用来赋值(例如，$x=3$)。两个等号用来定义一个条件。另一个例子是：

```
if x>=5 & x ~=10
```

则只有当 x 大于或等于 5，且不等于 10 时，才会执行 if 后面的语句。

表 4-1 MATLAB 中常用的关系和逻辑运算符

关系运算符	
==	等于
<	小于
>	大于
<=	小于等于
>=	大于等于
~=	不等于
逻辑运算符	
&	与
|	或
~	非

分析下面这个例子，修改 iftest 脚本，增加更多的逻辑判断。

```
1 % Get the user input
2 y = input('Enter a number between 1 and 10: ');
3
4
5 % If the user entered a number outside the range, change it to 10
6 if y > 10 | y < 1
7  fprintf('The number you entered is outside the range. It will be changed
   to 10\n')
8  y = 10;
9 end;
10
11 % Display the number
12 y
```

注意，我们用逻辑运算符“或”(第 6 行的，“|”符号，按下 Shift+\组合键可以输入这个符号)增加了多个逻辑判断。测试这个程序 3 次，分别输入 0、4 和 11。

我们还可以在分支里嵌入逻辑判断，这样得到更加复杂的逻辑结构。进一步修改这个脚本文件，把它改为如下形式：

```
1 % Get the user input
2 y = input('Enter a number between 1 and 10: ');
3
4
5 % If the user entered a number outside the range, change it appropriately
6 if y > 10 | y < 1
7  fprintf('The number you entered is outside the range. It will be
   changed.\n')
8      if y > 10
9          y = 10;
10         fprintf('The number has been changed to 10.\n');
11  end;
12  if y < 1
13   y = 1;
14   fprintf('The number has been changed to 1.\n');
15     end;
16 end;
17
18 % Display the number
19 y
```

测试这个程序 3 次，分别输入 0、4 和 11，看看它运行的结果。

图 4-14 是这个算法的流程图。虽然，这里的逻辑结构比前一个例子要复杂许多，但是逻辑语句采用的仍然是基本的 if 语句。每个 if 语句后面紧跟一个或多个只有当条件成立时才执行的命令，然后程序继续执行 end 之后的语句。注意在流程图中，两个路径连接处使用一个连接器。每个连接器对应于 MATLAB 里的一个 end 语句。

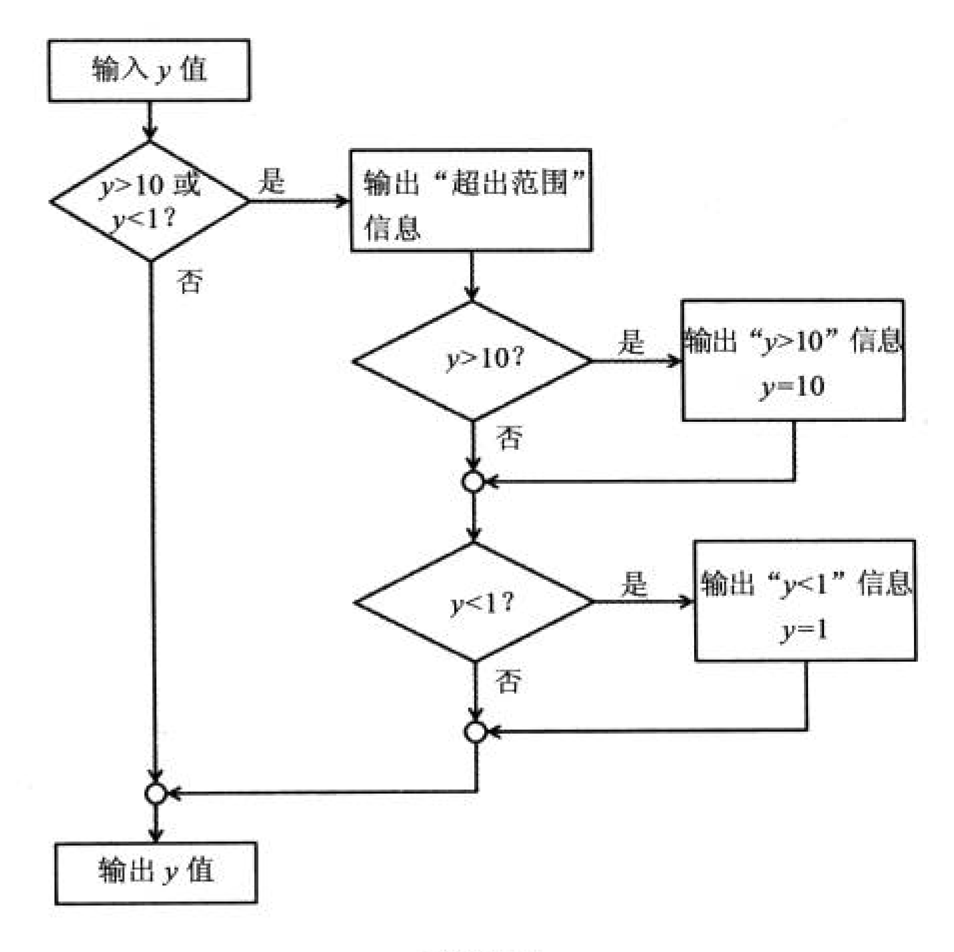


图 4-14

到现在为止，读者可能已经注意到，MATLAB 里的 if 语句好像有点不同于 Excel 里的 if 语句。在 Excel 里，我们知道，条件成立时，它将返回一个值，条件不成立时，将返

回另一个值。在 MATLAB 里，当条件成立时，if 语句确实要执行某些命令，但是当条件不成立时，不执行任何命令。在 MATLAB 里与 Excel 中的 if 语句相似的结构是 if-else 结构。下一小节将讨论 if-else 结构和 if-elseif-else 结构。

4.3.2 添加 else 和 elseif 条件

有时，我们执行逻辑分支时，希望在条件成立时执行其中一个分支，当条件不成立时，执行另一个分支。这一类的功能，在前一个例子里，要用两个不同的 if 语句来实现。现在可以用 if-else 模块来实现，而且显得更加简单。可以这样来理解 if-else 模块：

"当条件成立时，执行某一块的代码，否则(else)执行另一块的代码。"

考虑前面的例子，它的流程结构如图 4-14 所示。注意，如果 y 的值在 1~10 之间，程序直接跳到 end 语句，并显示 y 的值。假如需要显示在这个范围内输入的值，可以按如图 4-15 所示进行修改。注意流程中的第一个决策点。不仅在"成立"(yes)的分支上有操作处理代码，而且在"不成立"的分支上也有处理步骤。

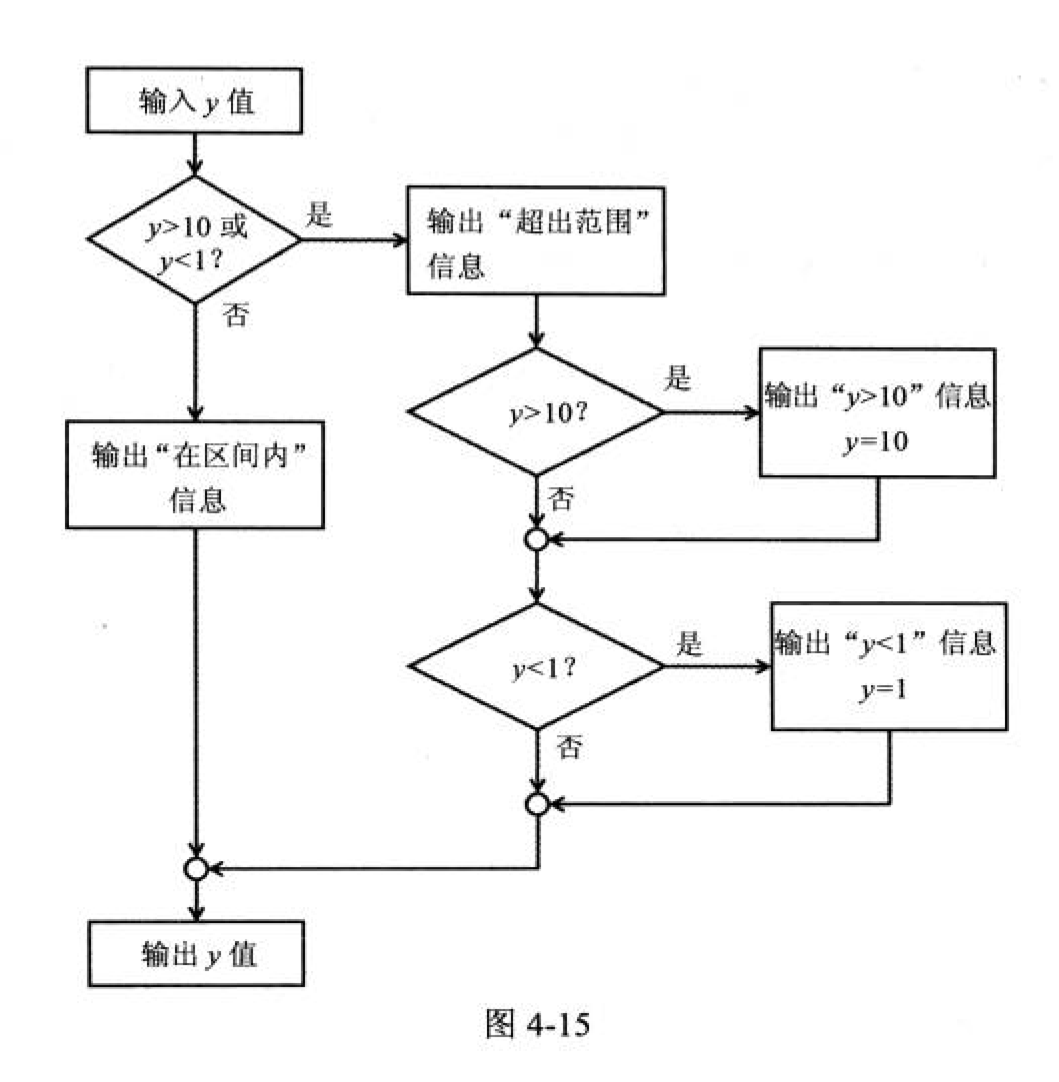


图 4-15

修改 iftest 脚本，如下所示(前面 15 行的内容没有变化)：

```
1  % Get the user input
2 y = input('Enter a number between 1 and 10: ');
3
4
5 % If the user entered a number outside the range, change it appropriately
6 if y > 10 | y < 1
7  fprintf('The number you entered is outside the range. It will be changed.
   \n')
8  if y > 10
9   y = 10;
10   fprintf('The number has been changed  to 10.\n');
11  end;
12  if y < 1
13   y = 1;
14   fprintf('The number has been changed to 1.\n');
15  end;
16 else
17  fprintf('The number is in the range\n')
18 end;
19
20 % Display the number
21 y
```

测试这个程序 3 次，分别输入 0、4 和 11。

注意，只有当第 6 行的 if 语句的逻辑条件为假时，才会执行 else 后面的语句(第 17 行)。

有了 if-else 模块，程序可以按两个不同的分支运行，但是实际上我们经常遇到多于两个分支的情形。此时，我们可以利用“梯形”结构把多个逻辑判断组合起来，即在 if 模块里添加一个或多个 elseif 语句。

例如，我们用 if-elseif-else 结构重写前面脚本里的逻辑结构。

```
1 % Get the user input
2 y = input('Enter a number between 1 and 10: ');
3
4
5 % If the user entered a number outside the range, change it appropriately
6 if y > 10
7  fprintf('The number is too high. It will be changed to 10\n')
8  y = 10;
9 elseif y < 1
10  fprintf('The number is too low. It will be changed to 1\n')
11  y = 1;
12 else
13  fprintf('The number is in the range\n')
14 end
15
16 % Display the number
17 y
```

同样，用 0、4 和 11 测试这个脚本。注意，这个脚本的逻辑关系与前面一个例子相同，但是这里的脚本可读性比较好，也比较简单。而且对应的流程图(见图 4-16)也比较简单和容易理解。

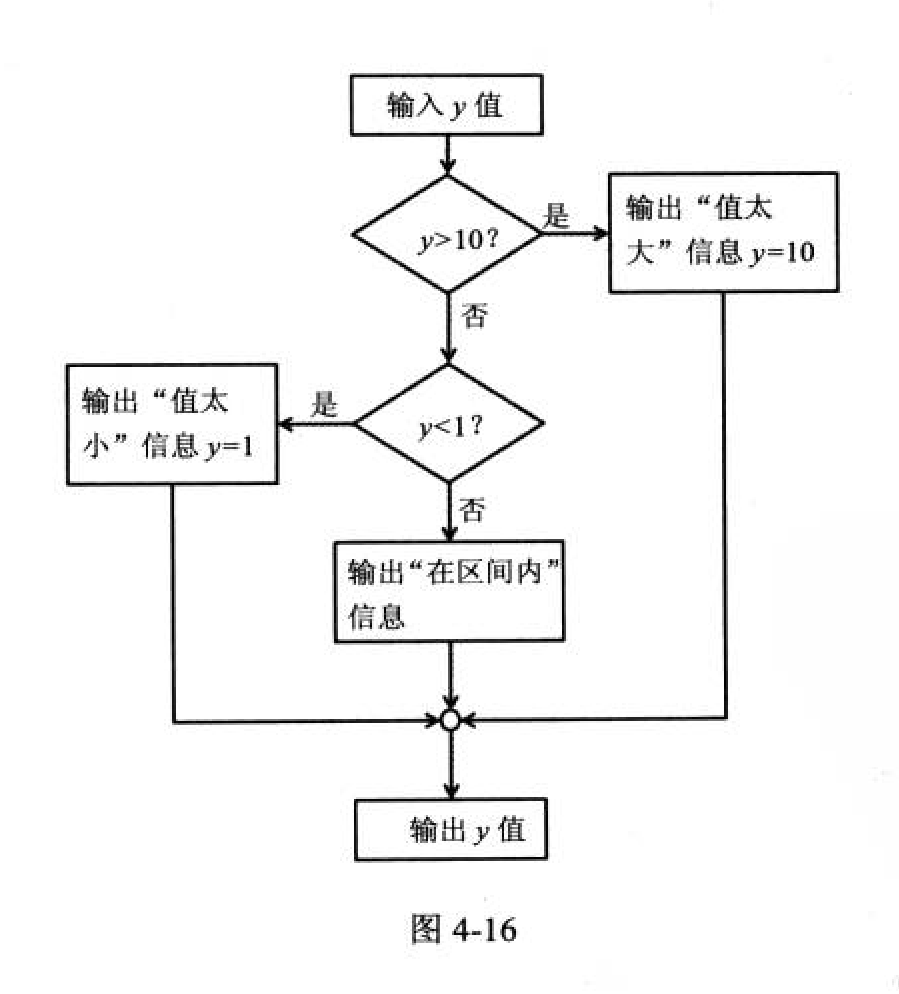


图 4-16

当我们掌握了这种逻辑分支的设计方法以后，就可以利用这种“梯形”结构，把更加复杂的逻辑分支组合起来。

例如，我们给前面的脚本增加更多的逻辑分支，如下所示：

```
1 % Get the user input
2 y = input('Enter a number between 1 and 10: ');
3
4
5 % If the user entered a number outside the range, change it appropriately
  and output the number
6 if y > 15
7  fprintf('The number is much too high. This program will terminate\n. ')
8 elseif y > 10
9  fprintf('The number is a little too high.It will be changed to 10\n')
10  y = 10
11 elseif y == 10
12  fprintf('The number is right at the upper limit\n')
13      y
14 elseif y <1
15  fprintf('Your number is too low. It will be changed to 1\n')
16      y = 1
17 else
18  fprintf('The number is in the range\n')
19  y
20 end;
```

输入 0、4、10、11 和 16 测试这个脚本。

深入分析输入 y=16 这种情况。虽然 y=16 会使第一个判断条件(y>15)和第二个判断条件(y>10)都满足，但是程序只会执行第一个分支(即 y>15 成立的分支)。在 if-elseif-else 结构中，逻辑判断是按顺序进行的。当其中一个条件成立时，之后的逻辑判断不会再进行测试。当程序执行了第一个判断条件成立的语句后，立刻执行 end 语句之后的代码。这个过程可以用图 4-17 的流程图来说明。第一个决策点对应于 if 语句。之后每个决策点对应每个 elseif 语句。在每个决策点位置，当条件成立时，程序流从分支(yes 分支)流出。如果条件不成立，则继续流向下一个决策点。这个过程反复执行，直到最后一个决策点为止。在最后一个决策点时，如果条件成立，也会从 yes 分支往下执行，但是，即使所有的条件都不满足(属 else 情形)，程序也会执行一段代码。所有的分支都在 end 语句重新合并成一个路径。

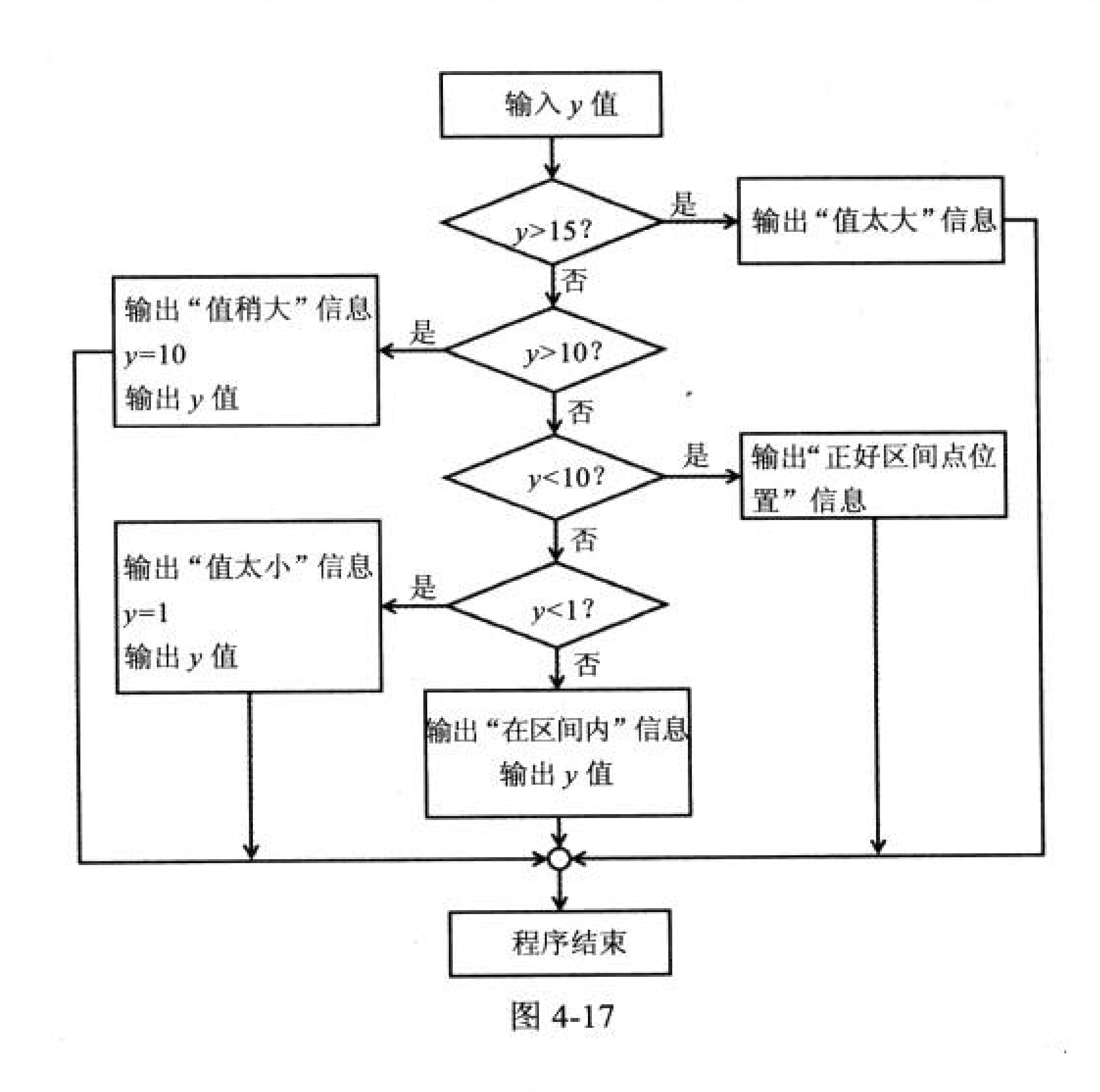


图 4-17

4.4 教程：循环结构和选择结构的综合应用

利用前面两节中学到的程序结构(循环结构和选择结构)，可以组成非常复杂、功能非常强大的程序。在本教程里，我们考虑两个例子，这两个例子都要求综合应用循环结构和选择结构。

例 4.1

在同一个坐标里绘制正弦波和方波图形。x 轴的范围为 0~4π。在数字电路中经常用到方波。与正弦波一样，方波也是一个周期函数，但是它从一个值(+1)到另一个值(－1)的跳变是瞬间完成的。方波的一个定义方法是把它与同周期的正弦波进行比较。当正弦波为正值时，方波函数值取+1，当正弦波为负值时，方波函数值为－1。

解：

本例需要一个 for 循环计算并保存绘制点的值。自变量 x 取 1000 个步长。因此，每次循环，x 的增量是 4π/1000。循环计数器 i 从 1 开始，每次循环它的增量为 1，终止值为 1001。循环中需要计算两个因变量的值，并把它们保存在 ysine 和 ysquare 两个数组里。

程序的流程图如图 4-18 所示。对于每个 x 值，我们先计算正弦波的值，并把它保存在 ysine 数组里。然后根据 ysine 的符号 ，计算 ysquare 的值。由于 ysquare 有 3 种可能的值，因此，使用 if-elseif-else 逻辑结构，效率会更高。

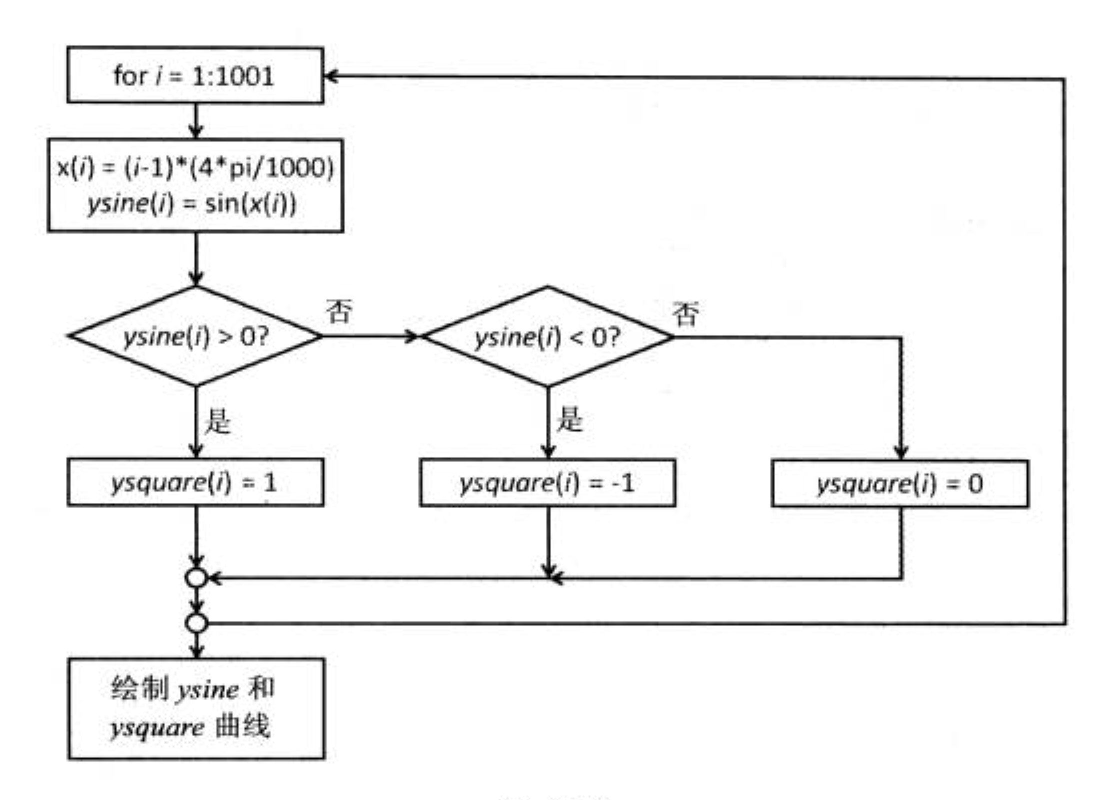


图 4-18

在新建的 m-file 里输入以下脚本：

```
1 %Clear Variables
2 clear;
3
4 %Wave Generation Loop
5 for i = 1:1001;
6       %Create Sine Wave
7       x(i) = (i-1)*(4*pi/1000);
8       ysine(i) = sin(x(i));
9       %Create Square Wave
10      if ysine(i) > 0;
11           ysquare(i) = 1;
12      elseif ysine(i) < 0;
13           ysquare(i) = -1;
14      else
15           ysquare(i) = 0;
16      end;
17 end;
18
19 %Plot data
20 hold off;
21 plot(x,ysine);
22 hold on;
23 plot(x,ysquare,'r');
24 axis([0 max(x) -1.2 1.2]);
```

保存并执行这个脚本程序。仔细分析，如何用 if-elseif-else 测试 ysine 的值，如何利用分支求 ysquare 的值。另外还要注意，程序中增加了几个图形格式设置命令。方波的画线的颜色是红色，因为它的 plot 命令使用 r 参数。axis 命令利用 max( )函数把图形的坐标轴设置为合适的缩放比例，如图 4-19 所示。注意，第 22 行 hold on 命令的作用是将这两个函

数的图形绘制在同一个图里。这个 hold on，表示保留当前的图形，因此可以在当前图形的基础上再绘制图形。在第 5 章中，我们将学习更多的图形绘制选项。

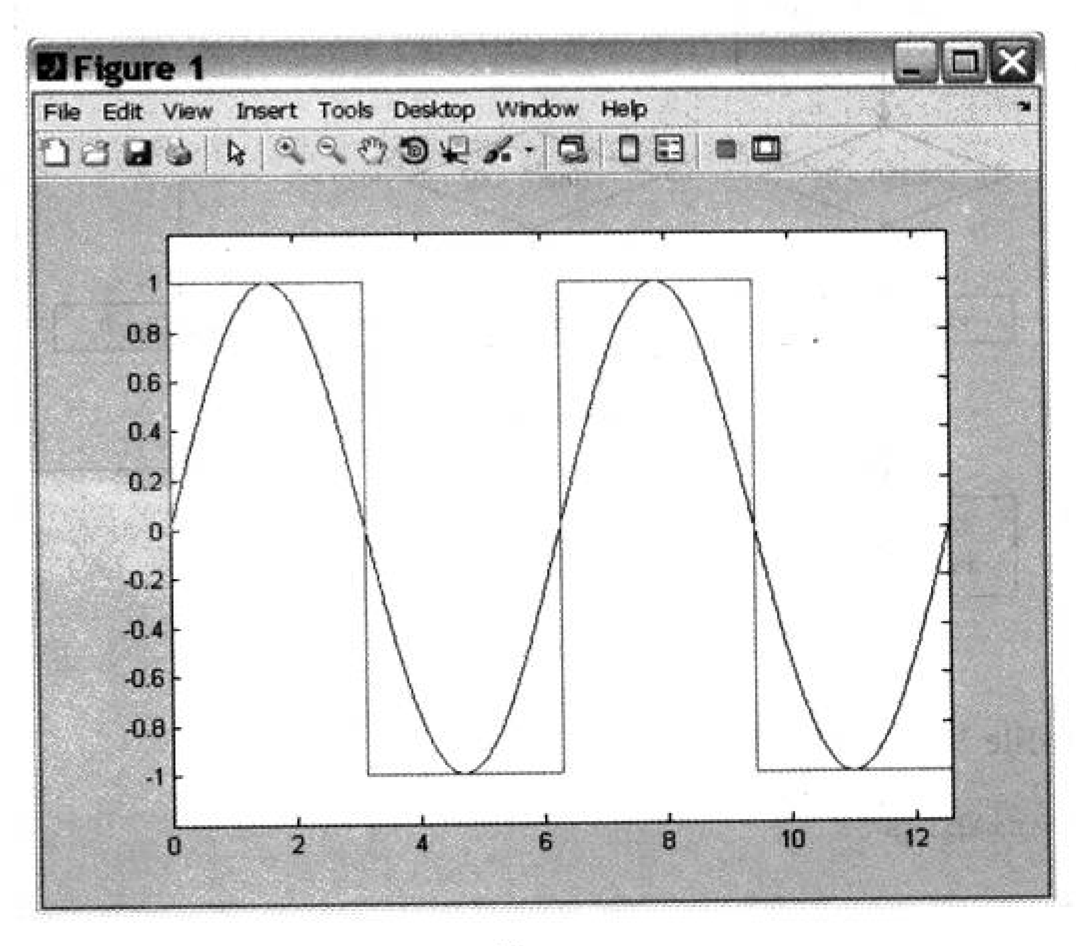


图 4-19

例 4.2

在许多教科书的习题里，特别是力学的教材里，经常遇到 3-4-5 三角形。3-4-5 三角形如图 4-20 所示。它可以简化许多运算。显然，这个三角形是一个直角三角形，角度 θ 的余弦值是 4/5，正弦值是 3/5，正切值是 3/4。

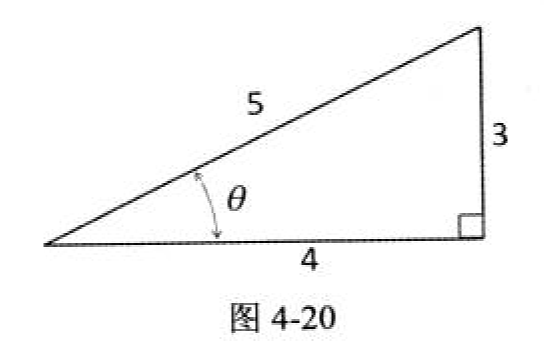


图 4-20

这个三角形的一个重要性质是两条短边的平方和是一个整数，而且是一个完全平方数，即它是另一个整数的平方。因此这个三角形的斜边是一个整数。

我们希望求得更多像这个三角形这样，边长都是整数的三角形。用枚举法，对每个三角形进行测试，找出所有边长在 25 以下，符合这个要求的三角形。

解：

用 x 和 y 表示这个三角形两条较短的边长。本程序脚本需要两个循环嵌套，如图 4-21 所示。

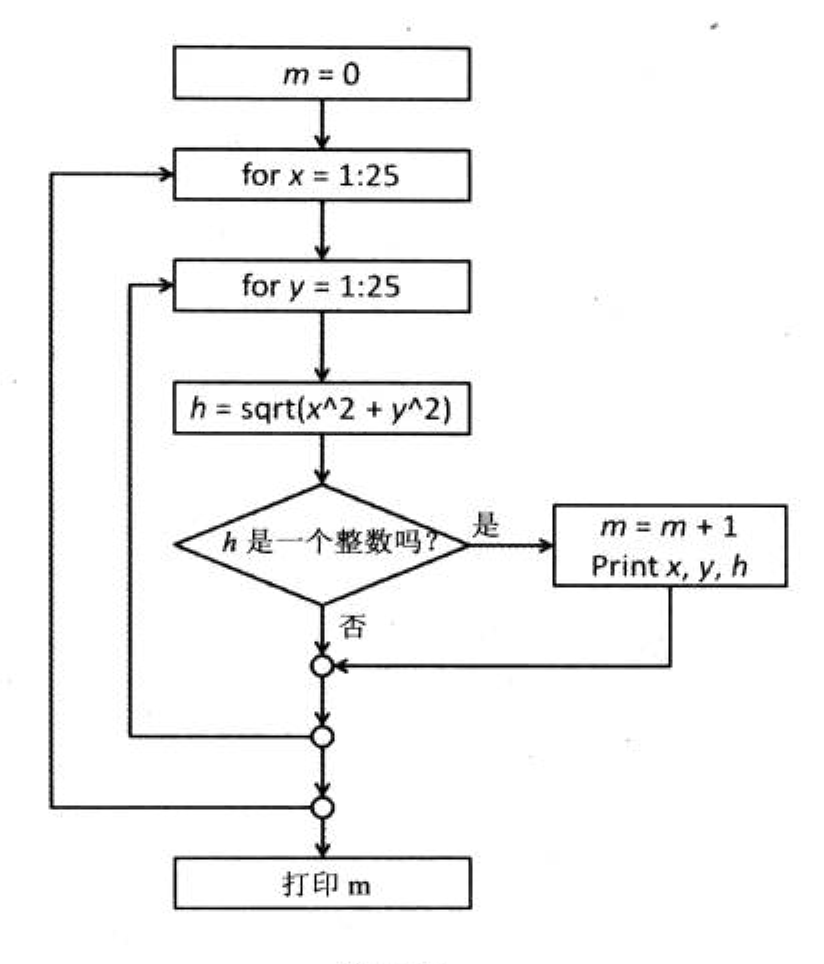


图 4-21

外循环让 x 的值从 1 递增到 25。对于每个 x 的值，我们把 y 的值从 1 递增到 25。因此，整个程序需要计算 x 和 y 的 25×25(即 625)种可能情况。但是我们如何知道每个三角形的斜边是否是一个整数呢？分析 MATLAB 的 floor( )函数。在命令窗口里输入 help floor 命令，可以得到这个函数用法的详细说明：

```
>> help floor
FLOOR Round towards minus infinity.
    FLOOR(X) rounds the elements of X to the nearest integers
    towards minus infinity.
```

floor( )函数返回不大于参数的最大整数。

正如读者所预料的那样，MATLAB 还有一个上限函数 ceil( )。它把一个数舍入到比它大的最接近整数。如果斜边 h 的值是一个整数，则 floor( )函数对它进行舍入运算不会对它的值有任何影响。因此，如果 h 等于 floor(h)，则 h 是一个整数。

在一个新建的 m-file 里，输入以下命令：

```
1    % This program searches for right triangles for
2    % which all three sides are integers
3
4    % Initialize the counter m
5    m = 0;
6
7    % Loop to check all combinations of x and y up to limits
8
9    for x = 1:25
10       for y = 1:25
```

```
11
12          % Calculate the hypotenuse h
13
14          h = sqrt(x^2 + y^2);
15
16          % Check to see if h is an integer
17          % If it is, write x, y, and h to the screen
18          % and advance the value of the counter m
19
20          if h == floor(h)
21             m = m + 1;
22             x
23             y
24             h
25          end
26      end
27  end
28  m
```

把这个脚本保存为 Triangle，并执行这个程序。

这个程序找出 22 个边长为整数的三角形。从 3-4-5 开始，接着是 4-3-5，5-12-13，6-8-10，7-24-25，等。我们在屏幕上看到的输出结果并不是十分一目了然。在下一节将学习如何对 MATLAB 输出结果进行格式设置。在这之前，读者可能已经注意到，第二个三角形 4-3-5，实质上与第一个三角形 3-4-5 属于同一个三角形。只是 x 与 y 的值交换位置。由于 x 和 y 的位置无关紧要，因此我们得到的结果是实际结果的两倍。

如何去掉这些重复的三角形呢？分析 for 循环里的 x 与 y 变量的变化。最初，我们把 x 设置为 1，让 y 从 1 递增到 25：

x=1

y=1,2,3,…,25

外循环的下次执行，x 是 2，y 又从 1 开始，递增到 25：

x=2

y=1,2,3,…,25

但是一个变量是 1，另一个变量为 2 这种情况已经计算过，因此，只要内循环从 2 开始，就可以去掉重复的结果。

外循环的第三次计算时，x 是 3：

x=3

y=1,2,3,…,25

现在，有两种情况已经计算过，即(1,3)和(2,3)。因此，内循环应该从 3 开始。

把这个分析推广到一般情形。内循环 y 的初始值必须等于 x 再开始计算。在 MATLAB 脚本程序里，这很容易实现。我们只要把第 10 行的内循环改为如下的语句：

```
for y=x:25
```

即把第 10 行的 1 改为 x，如前所示。保存并执行这个修改后的脚本程序。

最后得到的 m 值，即找到的三角形个数是 11，是原来的 22 的一半。

不要关闭这个脚本程序，下面的输出结果格式设置要用到这个脚本。

4.5 教程：MATLAB 输出结果的格式设置

在本章的前面几节里，我们用 fprintf( )函数把文本消息输出到屏幕上。在这一节，读者将要学习用 fprintf( )函数把变量的值输出到屏幕上或一个结果文件里，还要学习如何对这些变量的值和相应的文字说明进行格式设置。

fprintf( )函数的用法是：

fprintf(fid, 'Text string including conversion specifications of variables', list of variables)

fid 是文件标识符，当结果输出到屏幕时，可以省略 fid。单引号之间的内容是每个变量的输出格式转换说明。每个输出格式说明以%开始，后面紧跟与该变量有关的字符串(可选)、小数位数和格式类型。格式类型有：

f=定点小数表示
e 或 E=科学表示法
i=整数表示法

例如，%8.1e 表示以科学表示法输出该变量的值，整个值占用 8 个字符，小数位数取 1 位。

字符串可以包含像\n 这样的特殊字符，它表示换到下一行。还有其他几个特殊字符，如\t 表示 tab。在 MATLAB 的帮助文档中搜索 fprintf( )的用法，就会看到有关 fprintf( )函数的各种格式选项。对于大多数应用，本节介绍的这些选项已经足够了。

为了说明 fprintf( )函数的用法。我们分析表 4-2 中所示的例子。在这些例子里，m=12，d=7532.1234。

在前一节的 Triangle 脚本程序里，每个三角形由边长 x、y 和 h 确定。每找到一个三角形，我们都希望把三边的长度显示在同一行里。

在 Triangle 脚本里，找到在屏幕上输出 x、y 和 h 的行，用一个 fprintf( )函数把它们输出到同一行里。此外，修改最后一行，这一行输出计数器 m 的值。其他值都不变。

```
20 if h == floor(h)
21 m = m + 1;
22 fprintf('%5i%5i%5i\n',x,y,h)
23 end
24 end
25 end
26 fprintf('\nThe number of triangles found =%i\n',m)
```

表 4-2 fprintf 命令的示例

命令	屏幕上的输出/说明
fprintf('The value of m is %i',m)	`The value of m is 12>>` 使用整数格式；注意，如果 fprintf( )中没有使用换行符，则下一个命令的提示符也将出现在同一行里

fprintf('The value of m is %i\n',m)	`The value of m is 12`
	`>>`
	字符串里的\n 会把提示符输出到下一行的行首

fprintf('The value of d is %f\n',d)	`The value of d is 7532.123400`
	`>>`
	定点数输出格式，但没有说明小数位数，这会按 MATLAB 的短格式输出全部位数

fprintf('The value of d is %.1f\n',d)	`The value of d is 7532.1`
	`>>`
	定点数格式，一位小数位数

fprintf('The value of d is %10.1f\n',d)	`The value of d is    7532.1`
	`>>`
	d 输出占 10 个字符宽度，经常利用这种输出格式对齐同一列中的数据

fprintf('The value of d is %.2e\n',d)	`The value of d is 7.53e+003`
	`>>`
	科学表示法，输出有两位小数

fprintf('The value of d is %.1f\n
The value of m is %i\n',d,m)
	`The value of d is 7532.1`
	`The value of m is 12`
	`>>`
	如果有多个变量需要输出，则它们按格式转换的顺序输出

>> fprintf('The value of d is %i\n',d)	`The value of d is 7.532123e+003`
	`>>`
	既然 *d* 不是一个整数，而这里 *i* 格式不起作用，因此输出采用默认格式，即采用科学表示法和 MATLAB 的短格式

需要注意的是，第 22 行的“\n”字符的作用是，在输出每个三角形的 *x*、*y* 和 *h* 的值后，自动换行到下一行。在第 26 行里，字符串的开始有一个“\n”字符，这会使在最后一个三角形输出后，跳过一行，输出这个字符串。而字符串后面的“\n”，会使提示符显示在下一行的行首。

保存修改后的脚本程序，在命令窗口里，输入 clc 命令，清除屏幕，运行 Triangle 脚本程序，结果如图 4-22 所示。

```
Command Window
>> Triangle
     3     4     5
     5    12    13
     6     8    10
     7    24    25
     8    15    17
     9    12    15
    10    24    26
    12    16    20
    15    20    25
    18    24    30
    20    21    29

The number of triangles found = 11
>> |
```

图 4-22

我们很容易把屏幕上的输出结果复制到 Word 或 Excel 文档里。但是，如果在屏幕上输出大量的数据，我们有时候希望直接把数据保存到一个结果文件里。为此需要先定义文件名。这可以用 fopen 命令实现。定义一个文件名的方法如下：

fid=fopen('filename','attribute');

把 filename 这个文件的标示符赋给变量*fid*(使用其他变量也可以)表示。如果生成的是一个文本文件，最好在文件名后加上.txt 扩展名。这样，当我们在 Windows 里双击这个文件名，就可以用默认的文本编辑器(通常是记事本程序)打开这个结果文件。文件的属性(attribute)用 wt 表示，它表示写入(write)文本文件(text)。如果这个文件还不存在，执行这个命令后，就会建立这个文件。如果系统中已存在这个文件，则它的全部内容会被新写入的内容覆盖。如果我们希望在一个已经存在的文件的末尾添加内容，要使用 at 属性。

现在要把 Triangle 程序的结果保存到 output.txt 文件里，需要把这个程序修改成如下的结果：

```
1   %This program searches for right triangles for
2   % which all three sides are integers
3
4   % Initialize the counter m
5   m = 0;
6
7   % Loop to check all combinations of x and y up to limits
8
9   fid = fopen('output.txt', 'wt');
10  fprintf(fid,' x y h\n');
11  fprintf(fid,' === === ===\n');
12
```

```
13  for x = 1:25
14      for y = x:25
15
16          % Calculate the hypotenuse h
17
18          h = sqrt(x^2 + y^2);
19
20        % Check to see if h is an integer
21        % If it is, write x, y, and h to the screen
22        % and advance the value of the counter m
23
24        if h == floor(h)
25        m = m + 1;
26        fprintf(fid, '%5i%5i%5i\n',x,y,h);
27      end
28    end
29  end
30  fprintf('\nThe number of triangles found =%i\n',m);
```

第9行的fopen命令建立一个具有写入属性的文件。第10行和第11行用于写入列标题。因为列标题只需写入一次，因此必须放在循环之前。fprintf( )函数的第一个参数是*fid*，表示把变量输出到文件里，而非屏幕上。此外在第26行里，增加了*fid*标识符。注意当我们用fprintf( )函数把输出结果保存到文件时，最后要在行尾加一个分号。否则会在命令窗口显示写入到文件里的字符个数。第30行输出符合要求的三角形个数。这里的fprintf( )函数并没有*fid*这个标识符，因此，它会把输出结果显示在屏幕上，而不是结果文件里。

保存并运行这个脚本文件。

现在只有三角形的个数显示在屏幕上，如图4-23所示。

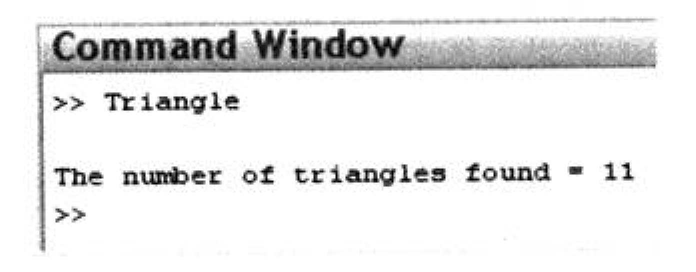


图4-23

我们注意到，在当前目录窗口里多了一个output.txt文件，如图4-24所示。这个文件可以用记事本、Word或Excel，甚至用MATLAB的编辑器打开。要想在MATLAB的编辑器里打开这个文件，只需要双击当前目录窗口的文件名。

这个文件的内容如图4-25所示。

All Files	Type	Size
intfun1.m	M-file	1 KB
normdist.m	M-file	1 KB
objfunc.m	M-file	1 KB
obpfunc.m	M-file	1 KB
output.txt	Text Document	1 KB
P1.m	M-file	1 KB

图 4-24

x	y	h
3	4	5
5	12	13
6	8	10
7	24	25
8	15	17
9	12	15
10	24	26
12	16	20
15	20	25
18	24	30
20	21	29

图 4-25

最后，把脚本中第 13 行和第 14 行的 x 和 y 的上限值修改为 1000，保存并运行这个文件。

在命令窗口里，我们看到有 456 个边长为整数的三角形符合要求。如果打开 output.txt 文件，就会在这个文件里看到全部的 456 个答案。

4.6 习题

1. 确定在运行以下每个 MATLAB 脚本之后 A 的值：

a.
```
A = 0;
for i = 1:5
    A = A + 5;
end
```

b.
```
A = 2;
while A < 10
    A = 2*A;
end
```

c.
```
A = 12;
if A > 10
    A = A/2;
end
if A >10
    A = A/2;
end
```

d.
```
for i = 1:5
    A(i) = i;
end
```

e.
```
A = 12;
if A < 10
   A = A*2;
else
   A = A/2;
end
```

f.
```
for m = 1:10
    A(m) = 5*m;
    if A(m) <= 25
       A(m) = 0;
    end
end
```

g.
```
for i = 1:3
    for j = 1:3
        if i == j
           A(i,j) = 1;
        else
           A(I,j) = 0;
        end
    end
end
```

2. 为习题 1 的每个脚本绘制流程图。

3. 下列每个 MATLAB 脚本都有错误，因此不能正常运行。指出其中的错误，并说明原因。

a.
```
for m = 1:5
    A = A+1;
end
```

b.
```
for i = 0:3
    A(i) = i+2;
end
```

c.
```
for k = 1:10
    A(i)= k;
End
```

d.
```
A = 4;
while A < 10
   A = A + 1;
   if A > 5
      A = 5;
   end
end
```

e.
```
for m = 1:1000
    j = m/10
    A(j) = m;
end
```

4. 假设某人在自己的账户里存入 1000 元，年利率为 5%。每年的年终，该账户里的总额是年初存入的金额的 1.05 倍。写一个 MATLAB 程序，用 for 循环计算 10 年、20 年、30 年后，该账户的本利和。

5. 对于习题 4。假设利息按年度计算，即每三个月把年利息的四分之一自动累加到这个账户里。假设利息按月计算，写一个 MATLAB 程序，用 for 循环计算 10 年、20 年、30 年后，该账户的本和利。

6. 对于习题 4 中的账户，写一个 MATLAB 程序，用 while 循环确定，多少年后，该账户的总金额会达到 5000 元。

7. 假设你从朋友那里借了 1000 元钱，他同意你按月分期还款，条件是要每月付未还款的 0.5%利息。建立一个 MATLAB 函数，它可计算并输出需要还款的月数。函数的输入参数是还款总金额。如果每月的还款为：

(a) 100 元

(b) 50 元

(c) 10 元

求出还完这笔借款需要多少个月？

如果你每个月的还款金额小于 5 元，结果会如何？

8. 在第 1 章里，我们讨论了炮弹飞行问题。当炮弹的发射角为 θ、初速度为 v 时，炮弹飞行的高度和水平距离与时间的关系如下：

$$h(t) = vt\sin\theta - \frac{1}{2}gt^2$$

$$x(t) = vt\cos\theta$$

编写一个MATLAB程序，当时间步长为 0.1s，发射角为 20°，初速度为 200ft/s，计算并保存炮弹在每个时间步长的飞行高度和水平距离。假设g=32.2ft/s²。一直计算到炮弹击中地面为止。用plot命令建立三个图形：

(a) 以 t 为水平轴，以 h 为垂直轴。

(b) 以 t 为水平轴，以 x 为重直轴。

(c) 以 x 为水平轴，以 h 为垂直轴(这样得到的图形就是炮弹的飞行轨迹)。

读者要么运行这个脚本程序三次，每次改变绘制需要的数据。或者在脚本里插入三个不同的 plot 命令。如果读者选择后一种方式，在 plot 命令之间要插入 figure 命令。figure 命令会打开一个新的绘图窗口，这样当前的 plot 命令不会覆盖前面的图形。

9. 在第 2 章的习题 13 里，我们建立一个电子表格计算标准纸张多次折叠的厚度。假设纸张的厚度为 0.004in。对折一次，厚度就变成 0.008，再对折一次，就变成 0.016，反复进行。写一个 MATLAB 程序，得到以下理论上的厚度需要对折多少次？

(a) 1in

(b) 1mile

(c) 从地球到月球的距离

10. 在材料力学的课程中，我们要学习如何计算横梁的偏移。利用不连续函数，我们用一个方程就可以描述多负载的横梁的偏移值。这个不连续方程的定义如下：

$$\langle x-a\rangle^n = \begin{cases} (x-a)^n & \text{当 } x \ge a \text{时} \\ 0 & \text{当} x<a \text{时} \end{cases}$$

对于如图 4-26 所示的钢架，横梁的偏移值 v 可以表示为：

$$v = \frac{1}{3.190\times10^9}\left(800x^3 - 13.68\times10^6 x - 2.5x^4 + 2.5\langle x-120\rangle^4 + 600\langle x-240\rangle^3\right)$$

式中，x 和 v 的单位都为 in。

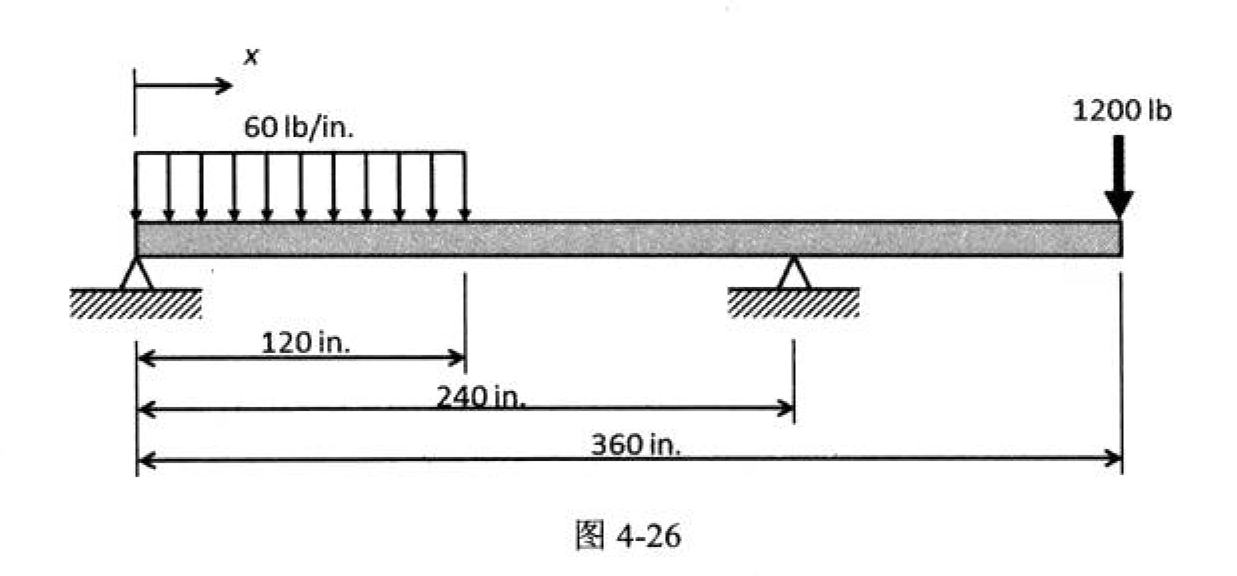


图 4-26

编写一个 MATLAB 程序，绘制一个图形反映横梁的偏移变化，并求最大偏移值距离左端的位置(以 in 为单位)。计算偏移值时取 x 的增量为 0.5in。

11. 再次回到第 4.4 节里的 Triangle 程序。当我们仔细分析输出结果时，发现许多三角形是相似的，如边长为 3-4-5 三角形与边长为 6-8-10 三角形。这两个三角形的形状是相似的，只是第二个三角形的边长都是第一个三角形的边长的两倍。修改这个脚本程序，不输出相似三角形。求有多少三角形符合此要求，假设三角形最短边的上限为 25。

提示：把符合要求的三角形的边长 x 和 y 值保存在数组里。当一个新的三角形符合要求时，我们先把它与已保存的三角形进行比较。如果新三角形的 x 值与已保存三角形的 x 值之比等于相应的 y 值之比，则新三角形的一个相似三角形已保存在数组里，因此这个新三角形的值不需要保存，也不需要输出到屏幕。

12. 房屋贷款通常的期限是 15 年或 30 年，每月还款。建立一个分期还款表是很有必要的，贷款人可以清楚看到每个月要付的利息和每个月的贷款余额。例如，一笔 20 万 30 年期限的贷款，年利率为 6.0%(还期为 360 个月，月利率为 0.5%)。根据银行利率计算公式，这样一笔贷款，每个月的还款额为 1199.10 元。编写一个 MATLAB 程序，建立分期还款表的文本文件。对于每个月，都要显示每月还贷前的贷款余额(是前一个月余额的 1.005 倍)和还贷额以及当月还贷后的余额。把这个分期还贷表设置为类似于图 4-27 的格式(这里只显示全部 360 行中的 12 行)。使用 while 循环，当还款余额小于 0 时，输出循环结果。在文件的结尾加上一行，由于多还贷，需要退还给贷款人的金额。如图 4-27 所示，注意，图中 1199.11 元还款额已经四舍五入到分，结果是经过 360 个月还贷后，总额稍稍多于贷款额。如果每个月的还贷额为 1199.10 元，则还贷 360 个月后，还有一笔小小的贷款余额没有偿还。

```
          Beginning                        Ending
Month      Balance        Payment          Balance
=====     =========       =======          =======
  1       201000.00       1199.11          199800.89
  2       200799.89       1199.11          199600.78
  3       200598.79       1199.11          199399.68
  4       200396.68       1199.11          199197.57
  5       200193.55       1199.11          198994.44
  6       199989.42       1199.11          198790.31
  7       199784.26       1199.11          198585.15
  8       199578.07       1199.11          198378.96
  9       199370.86       1199.11          198171.75
 10       199162.61       1199.11          197963.50
 11       198953.32       1199.11          197754.21
 12       198742.98       1199.11          197543.87


359        2383.31        1199.11            1184.20
360        1190.12        1199.11              -8.99

Final balance to be refunded =  8.99
```

图 4-27

13. 修改习题 12 中的 MATLAB 程序，把每个月的还贷额定为 1300 元，结果会如何？

14. 考虑习题 12 中的问题。假设从贷款的第 12 个月开始，借款人每年都会收到股票红利。把每年收到的红利 1000 元作为当月的额外还贷额。修改习题 12 的脚本程序，每隔 12 个月，他的还贷额增加为 2199.11 元，在其他月份，他的还贷额仍然还是 1199.11 元。这样还贷的结果如何？

第 5 章

数据的图形表示

引言

在进行工程计算时，我们经常会遇到大量的数据。有时这些数据可以用表格来表示。但是用图表表示数据也总是很有必要的。与一个大型表格相比，读者更容易理解一个精心设计的图表。图表经常可以帮助我们解释数据、显示因果关系。这是其他方法很难做到的。

本章我们将学习以下内容：

- 在工程文件中经常用到的不同图表类型。
- 用 Excel 和 MATLAB 建立函数和数据的散点图。
- 如何用曲线拟合数据。
- 如何设计容易理解的图表。
- 如何用 Excel 建立饼图、柱形图和组合图。

5.1 图表的类型

在介绍各种类型的图表之前，有必要说明一下，图表(graph)、图(chart)和绘图(plot)这 3 个词经常互换使用。在 Excel 里，所有的图表(graph)都可以被称为图(chart)。在 MATLAB 里，用 plot 命令建立图表。在工程中，最有应用价值的图表是 XY 图或散点图。在这些图中，数据对会被绘制在栅格上。通常，自变量作为水平 x 轴，因变量作为垂直 y 轴。但是在某些情况下，我们无法确定一个变量与另一个变量的关系。此时把数据绘制成图表可以帮助我们确定这两个变量是否存在某种关系。有时我们把 XY 图绘制成光滑的曲线，如图 5-1 的正弦曲线。有时，用同一个 x 坐标轴表示多个变量，如同时绘制角度的正弦曲线和余弦曲线，如图 5-2 所示。在这样的图表里，用图例可表示不同的曲线。

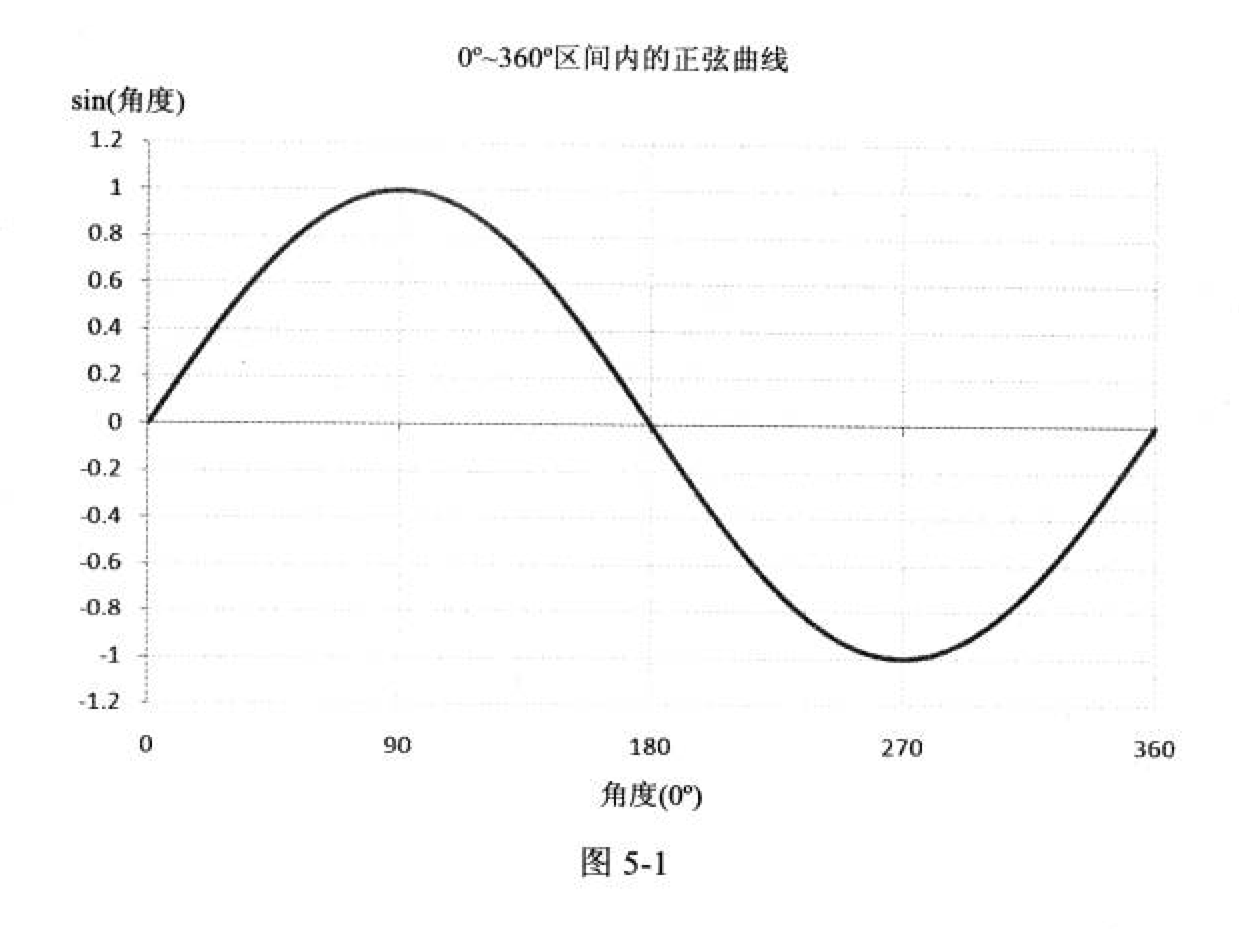


图 5-1

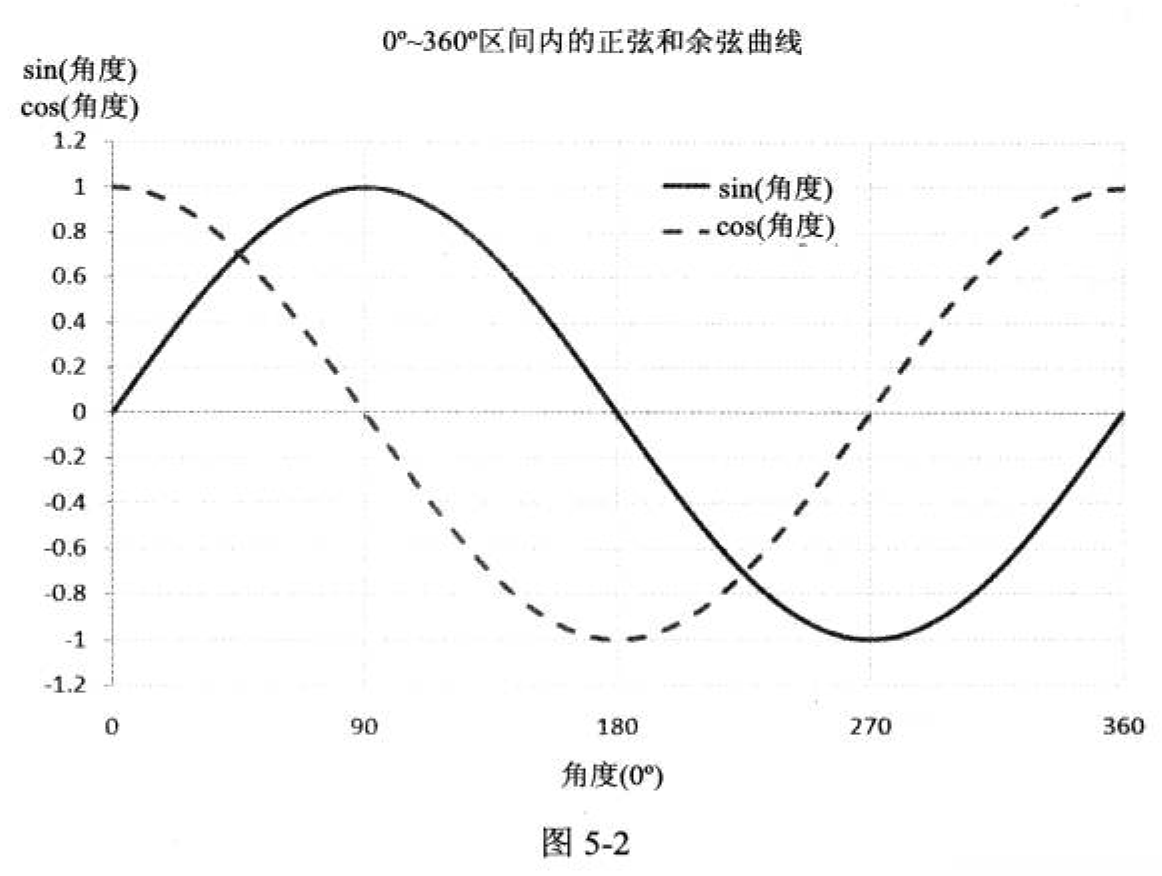


图 5-2

当在图上绘制实际的数据点时，我们经常用数据标志表示这些数据点，如图 5-3 所示。当出现很多这样的数据点时，就会给人一种“散乱”的感觉，这正是使用“散点图”这个名词的原因。但是，像这种没有用线段连接起来的 XY 图并不常见。在绘制数据点时，还经常绘制以下这些类型的曲线：

- 理论曲线，它显示这些数据点与理论值的符合程度。
- 拟合曲线(在 Excel 里被称为趋势曲线)，它是根据这些数据点值绘制得到的曲线。
- 由一系列直线段拼成的曲线。这种曲线是我们最不喜欢的，因为直线段不能表示任何方程。

在图 5-4 里，增加了理论曲线。在这个图里，数据点代表细钢条屈曲试验的轴向压力值。图中的光滑曲线代表理论的屈曲载荷，它是由材料力学的基本原理推导得到的方程的计算结果。在本科生的工程实验中，经常用这类比较图说明实验结果与理论值的符合程度。然而，当新理论需要实验数据验证时，这类图也经常出现在技术文献里。

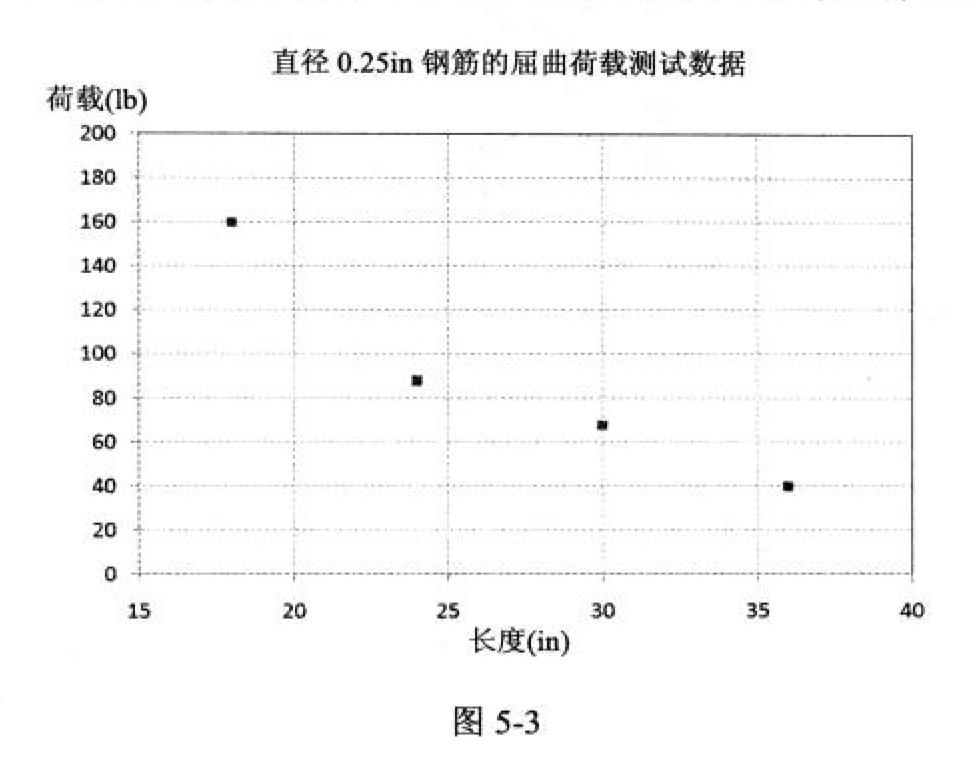


图 5-3

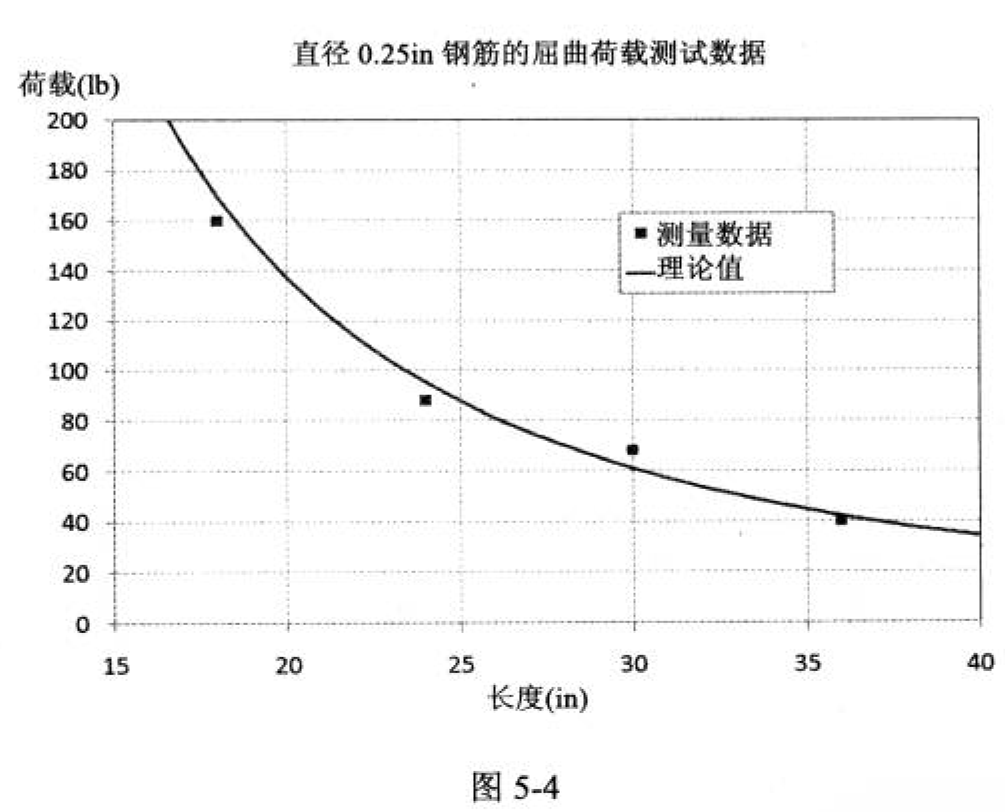


图 5-4

XY 图是工程报告和技术论文里最经常使用的图表，但是其他几类的图表也经常用来显示数据。其中最简单的图表是饼图和柱形图。这类图经常用来显示一维数据集中各项值的相对关系。饼图，顾名思义，就是把每一项数值显示为圆的一个扇区，它相当于馅饼的一部分。这类图的一个优点是，图形化表示一项值与另一项值的相对关系，或者部分值与整体(整个饼图)的相对关系。在图 5-5 里，用饼图表示一个班级考试成绩的分布情况。需要注意的是，根据这个饼图里各个扇形所占的面积，我们很快就大致知道它们之间的大小关系。如 A 等的人数是 F 等的两倍。C 等的人数占全班的三分之一左右。

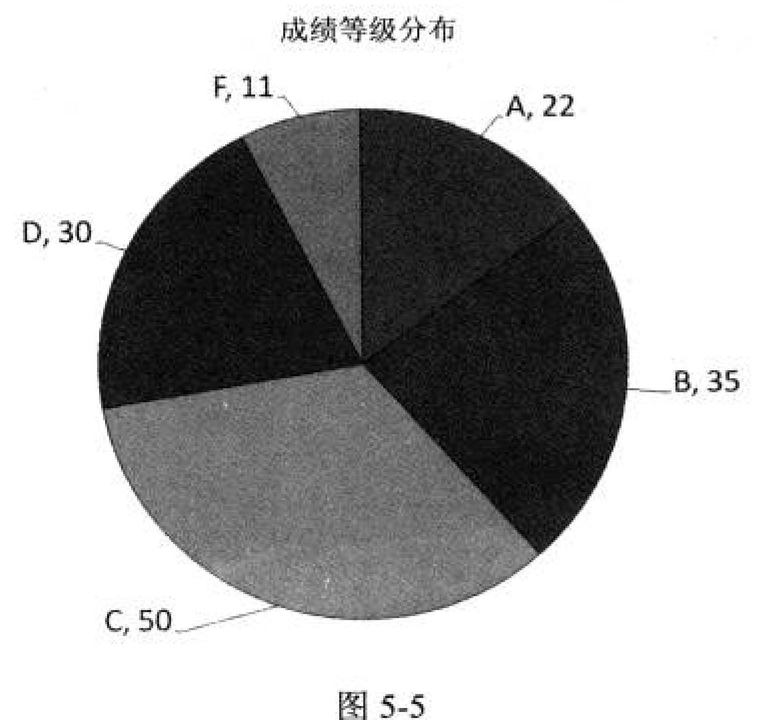


图 5-5

柱形图用矩形显示每项的数值，矩形的高度与数值成正比，如图 5-6 所示。注意，在 Excel 里，这类图又称为柱形图，另外一类相似的、矩形水平放置的图表称为条形图。在本书里，我们把这两类图统称为柱形图，不管图中的矩形是水平放置还是垂直放置。

柱形图的一个优点是，可以把两组甚至多组数据并列显示。例如图 5-7 同时显示第二次考试结果和第一次考试结果。这种并列显示数据组的方式是我们比较两组数据的很有效的方法。

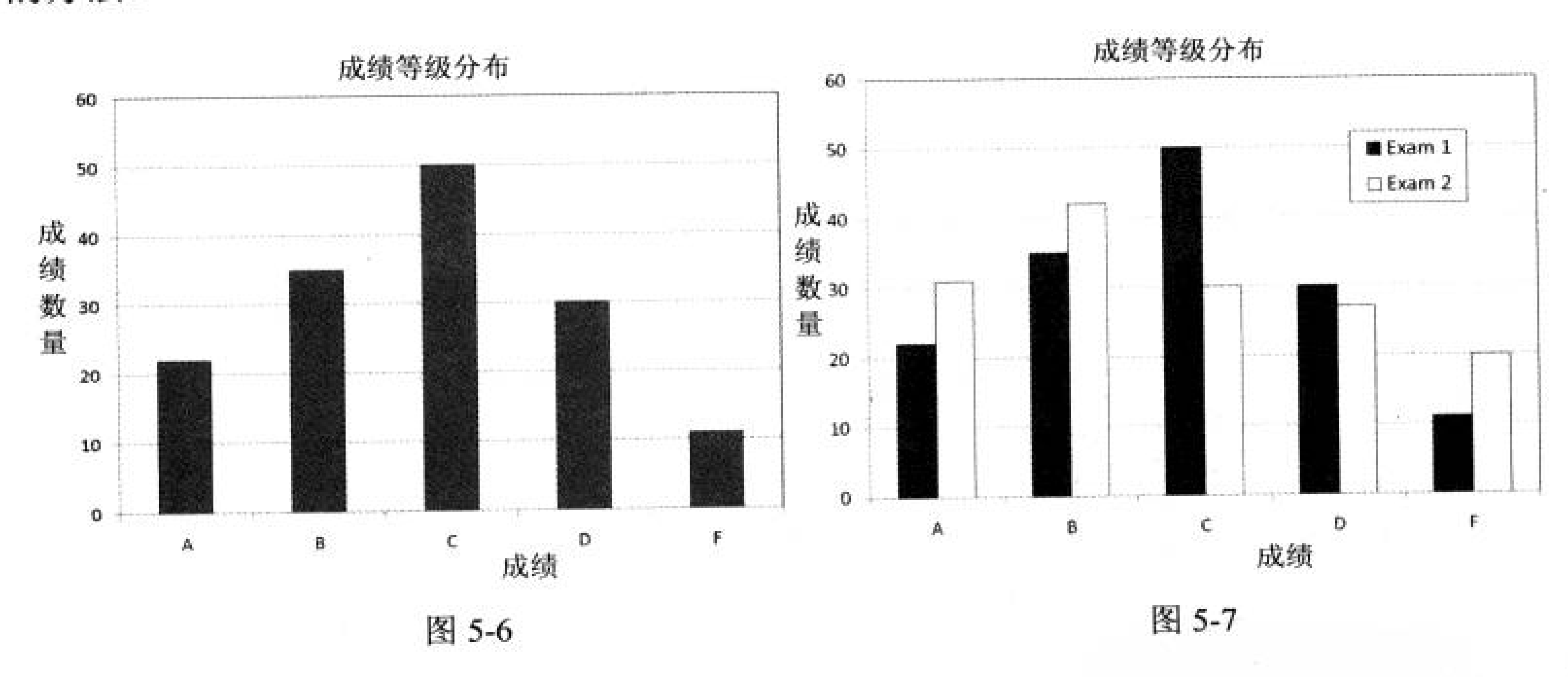


图 5-6　　图 5-7

有一类特殊的柱形图在统计分析中特别有用，它就是直方图。在直方图里，数据显示为数值范围，数值范围被称为箱体(bin)。例如，假设我们问一组学生，他们在作业上所花的时间。他们的回答是从 45 分钟到 8.5 个小时。简单的办法是取箱体的间隔为 1 小时，则第一个箱体包括了 1 小时和小于 1 小时的时间。第二个箱体包括了大于 1 小时、小于等于 2 小时范围内的时间。依此类推，最后的箱体包括了所有大于 8 小时的时间。根据这个结果，我们就可以建立一个柱形图，显示在每个时间间隔内的学生人数，如图 5-8 所示。注意，图中的各柱体之间没有间隔，这是因为水平轴的值是连续的，而在图 5-6 和图 5-7

里，水平轴上的值是离散的。

另一类在质量控制和过程控制中广泛使用的柱形图是 Pareto 图。在 Pareto 图里，数据按类型排列。例如，某个新型汽车模型在购买之后的最初几个月可能需要维修服务。每次上门维护的原因可以分类为引擎故障、油漆坏点、排气系统故障，等。每类问题的上门维修次数按照故障类型的顺序绘制成柱形图，把故障出现次数最多的排在最前面，把故障出现最少的排在最后面，如图 5-9 所示。Pareto 图经常包含累积百分比曲线。这个曲线显示了到某一类故障为止的上门维修次数的百分比。例如，根据图 5-9 可知，该曲线在传动故障(Transmission)位置的值为 80%。这意味着，分类为排气、电气和传动的上门服务占总服务的 80%。

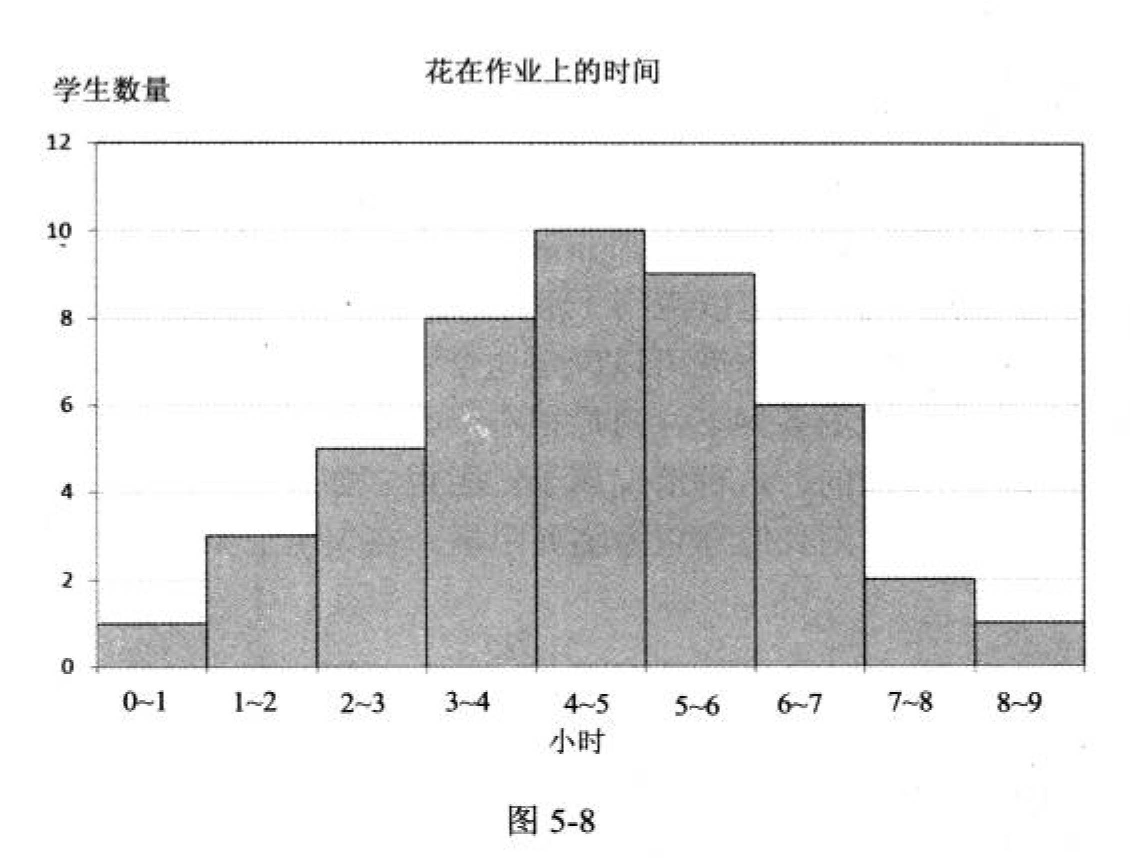


图 5-8

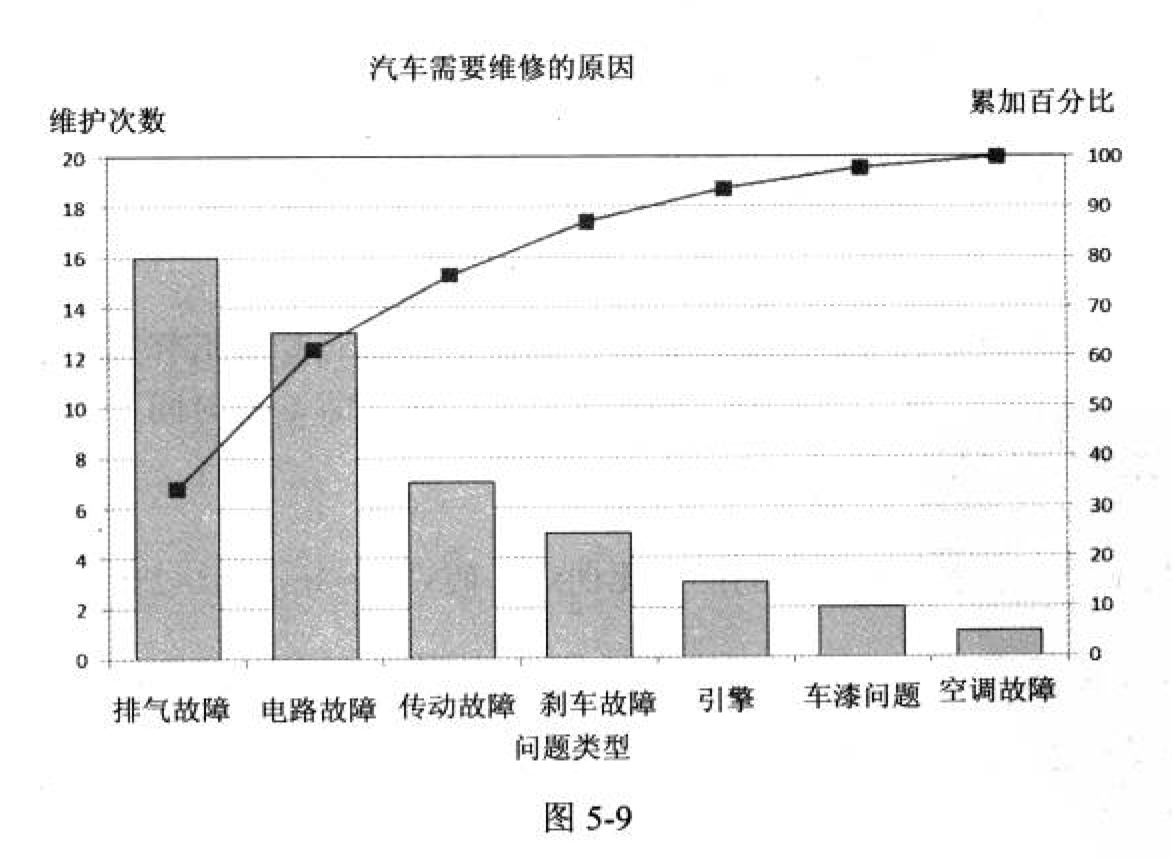


图 5-9

5.2 XY 图表

正如前面曾提到的，XY 图是工程中应用最广泛的一类图表。下面两个教程向读者介绍在 Excel 和 MATLAB 里建立 XY 图表的过程。

5.2.1 教程：在 Excel 里绘制方程

在本教程里，我们将绘制一个比较复杂方程的值。这个方程描述了一个弹簧-振子-阻尼系统的响应。假设在弹簧的一端挂着一个振子，我们用力把这个振子拉离平衡位置，然后放手，这个振子就做上下振动。最后，振动逐渐衰减，直到我们不能觉察为止。振动的衰减快慢是系统的阻尼系数的函数。在弹簧振子这样最简单的系统里，阻尼系数很小。但是在其他一些系统里，比如在汽车的避震系统里，阻尼器(减震器)是不可缺少的一部分。这个弹簧振子系统在工程的许多领域里都非常有用。弹簧振子模型可以应用于避震系统、建筑结构的振动等机械系统，也可以应用于液压系统。电子线路也存在类似的阻尼问题。

列出并求解该系统的微分方程就可以确定这个系统的响应。我们先分析在欠阻尼条件下方程的求解结果。在欠阻尼条件下，阻尼很小，允许振子振动。如果阻尼很大，则振子回到原来的位置都是不可能的，这种情况属于过阻尼。如果阻尼正好使得振子回到原来位置，但是并没有引起振动，则我们称它为临界阻尼。在欠阻尼条件下，弹簧振子的解是：

$$y=\left[y_0\cos\omega_D t+\frac{y_0\xi\omega}{\omega_D}\sin\omega_D t\right]e^{-\xi\omega t} \tag{5-1}$$

- y 是振子相对于平衡位置的偏移量。
- y_0 是初始偏移值(即在 t=0 时刻的偏移量)。
- ω 是系统的固有振动频率。它表示系统每秒钟自由振动的次数。
- ξ 是阻尼系数。它的值介于 0~1 之间。
- ω_D 是阻尼频率，它的计算公式是：

$$\omega_D=\omega\sqrt{1-\xi^2} \tag{5-2}$$

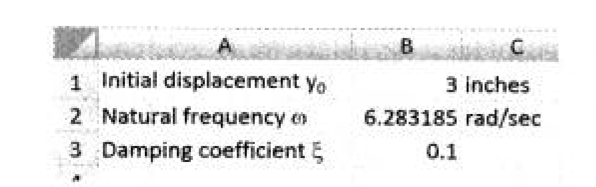
这个解是假设初速度为 0 时得到的，即把弹簧拉伸到 y_0 位置后释放。我们要求出因变量与自变量 t 的函数表达式，前绘制出它们的变化曲线。即 y 是时间 t 的函数，因为方程 5-1 中其他参数都是常量。

例 5.1

绘制弹簧振子系统的位移随时间的变化曲线。假设 y_0=3in，ω=2π rad/s，ξ=0.10。

解：

在 Excel 里新建一个工作簿。并在 3 个单元格里分别输入 y_0、ω 和 ξ 等符号，在相应的单元里输入前面给定的值，如图 5-10 所示。在固有频率(Natural Frequency)对应的单元格里输入公式“=2*PI( )”。



	A	B	C
1	Initial displacement y_0	3	inches
2	Natural frequency ω	6.283185	rad/sec
3	Damping coefficient ξ	0.1	

图 5-10

回忆一下，PI( )是 Excel 的一个函数，它返回π的值。由于 Excel 里的全部函数都要求在括号里输入一个参数，因此在 Excel 里，如果在函数后面跟一个不带任何参数的括号表示它是一个函数。

其实并没有必要在单元里输入ω等符号。但是如果读者确实想插入这些标号，其步骤是：从 Ribbon 中选择 Insert:Symbol 命令，从【字体】列表中选择 Symbol 选项，如图 5-11 所示，从中选择所需要的符号，单击对话框里的【插入】按钮，就把这个符号插入到单元的标题里了。读者可能想减少固有频率值显示的小数位数，但是如果读者确实想这样做，不要把它的小数位数设置为小于 3 位，因为我们要把它的值与阻尼频率进行比较，后面将介绍如何求阻尼频率。

下标的输入方法是：选择 Ribbon 的开始页，单击【字体】组右下角的下拉箭头，出现设置“单元格式”对话框，如图 5-12 所示。选择【下标】选项。

在 A5 单元里输入阻尼频率的符号，并在 B5 单元里输入计算阻尼的公式 5-2，如图 5-13 所示。



图 5-11

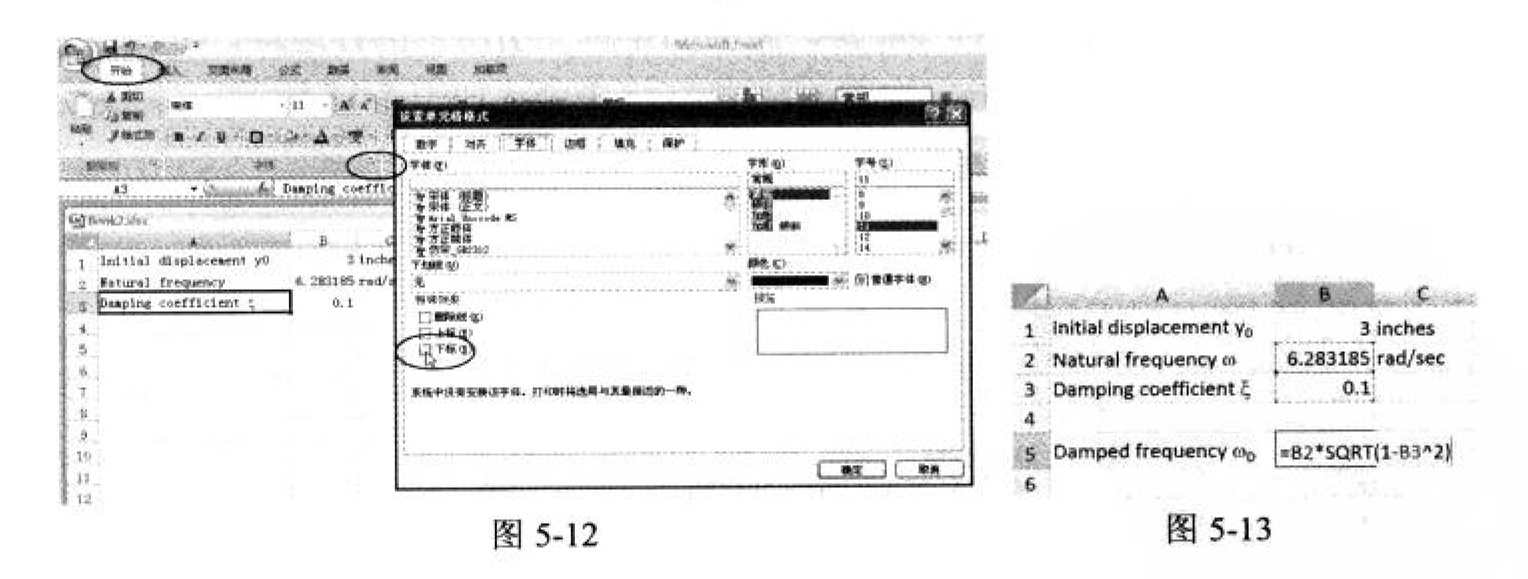


图 5-12　　　　图 5-13

在标题单元里输入ω_D，必须在输入了ω之后把字体切换到【常规】。

注意利用阻尼公式计算得到的值，它的值是 6.252rad/s，这个值非常接近系统的固有频率。当阻尼系数比较小时，固有频率与阻尼频率近似相等。

在时间和位移单元里输入标题。在时间一列里，输入 0、0.1、0.2 等，直到 1.0 为止，如图 5-14 所示。在单元 B8 里，输入方程 5-1 的 Excel 公式：

*“=(B1*COS(B5*A8)+B1*B3*B2/B5*SIN(B5*A8))*
EXP(-B3*B2*A8)”*.

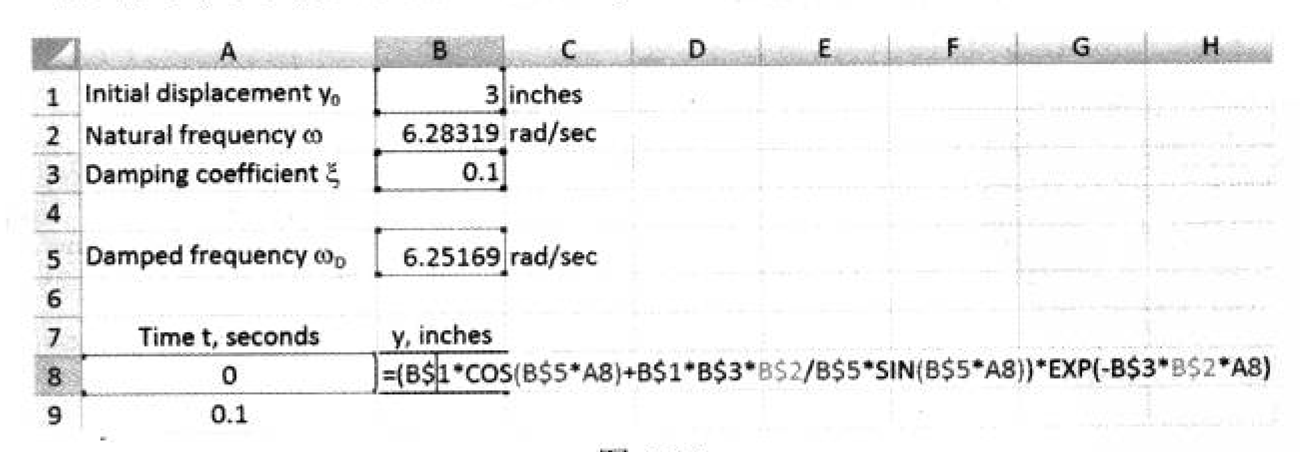
按下 Enter 键，结束公式的输入。返回的值应该是 3.0in，因为这正是初始位移值。

	A	B	C
1	Initial displacement y_0	3	inches
2	Natural frequency ω	6.28319	rad/sec
3	Damping coefficient ξ	0.1	
4			
5	Damped frequency ω_D	6.25169	rad/sec
6			
7	Time t, seconds	y, inches	
8	0		
9	0.1		
10	0.2		
11	0.3		
12	0.4		
13	0.5		
14	0.6		
15	0.7		
16	0.8		
17	0.9		
18	1		

图 5-14

在把这个单元里的公式复制到其他单元格之前，我们需要以绝对方式引用 B1:B5 单元，因为在复制过程中，它们的值需要保持不变。在本例里，我们只需要固定这些单元的行号就可以，不需要固定它们的列号。这样，当我们为了比较两组不同的输入值需要建立另一列时，可以很方便把这里的公式复制到另一列里。

双击单元 B8，修改其中的公式。对于 B1、B2、B3、B5 单元的引用地址，把光标移动到列号与行号之间，连续两次按 F4 键，就会固定单元引用中的行地址，因此会在行号之前插入一个$符号(列号前面没有这个符号)。现在这个公式如图 5-15 所示。



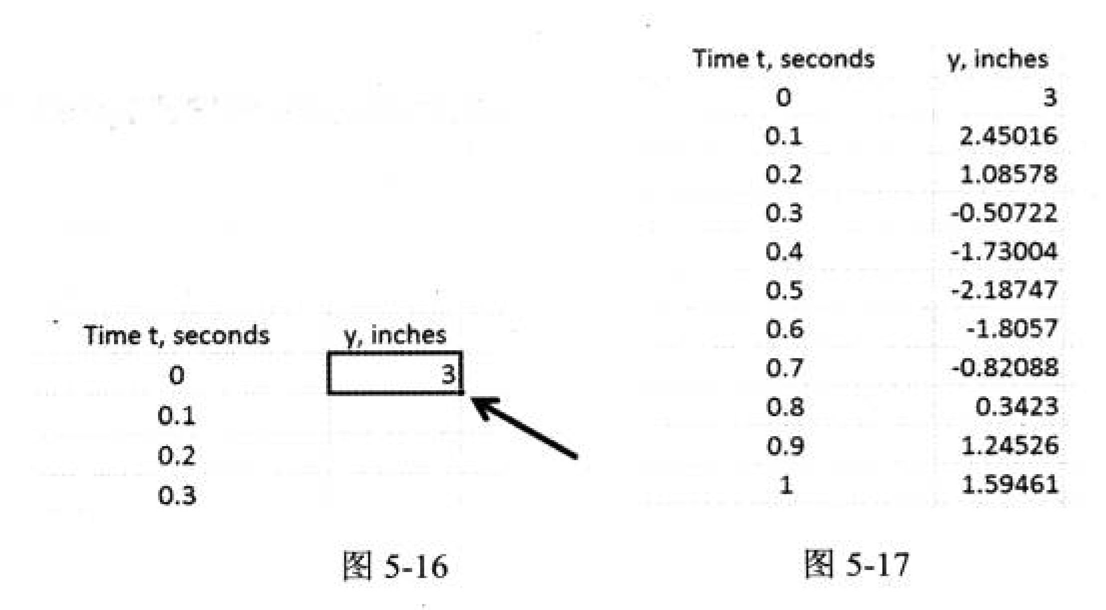
	A	B	C
1	Initial displacement y_0	3	inches
2	Natural frequency ω	6.28319	rad/sec
3	Damping coefficient ξ	0.1	
4			
5	Damped frequency ω_D	6.25169	rad/sec
6			
7	Time t, seconds	y, inches	
8	0	=(B$1*COS(B$5*A8)+B$1*B$3*B$2/B$5*SIN(B$5*A8))*EXP(-B$3*B$2*A8)	
9	0.1		

图 5-15

现在双击 B8 单元的右下角的填充柄，如图 5-16 所示，这会把该单元的公式复制到 B

列中 B8 单元之后的单元格。读者把自己的计算结果与图 5-17 进行比较。

选择 A8:B18 单元，在 Ribbon 的【插入】选项卡里，选择【散点】图表，并选择【带平滑线的散点图】，如图 5-18 所示。



Time t, seconds	y, inches
0	3
0.1	
0.2	
0.3	

图 5-16

Time t, seconds	y, inches
0	3
0.1	2.45016
0.2	1.08578
0.3	-0.50722
0.4	-1.73004
0.5	-2.18747
0.6	-1.8057
0.7	-0.82088
0.8	0.3423
0.9	1.24526
1	1.59461

图 5-17

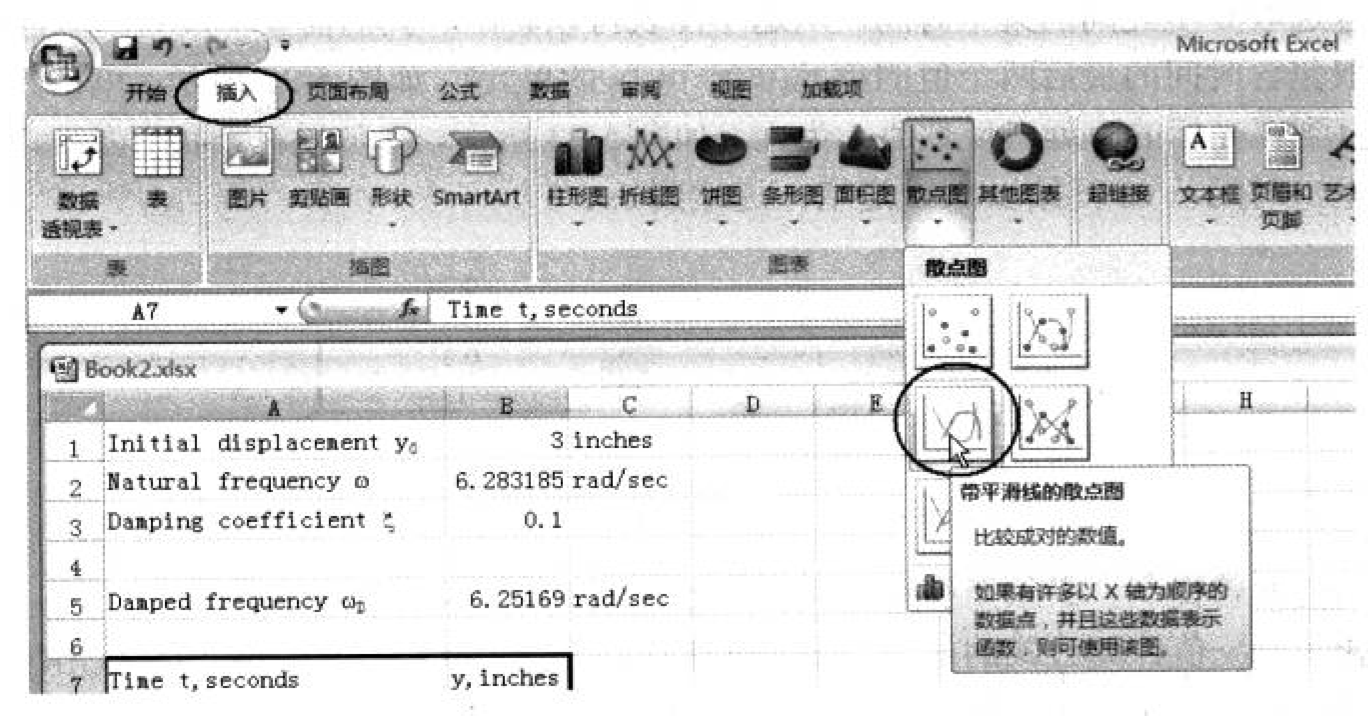


图 5-18

最后得到的图形如图 5-19 所示。在默认情况下，Excel 将把图表放在当前工作页上。但是，如果图表需要打印、复制或粘贴到另一个应用程序，则把图表放在一个单独页面里可以更好地控制它的外观。

需要注意的是，如果我们建立或选择了一个图表，Ribbon 会增加 3 个与图表有关的工具组：设计、布局和格式。

单击图表工具栏的【设计】选项卡，选择【移动图表】选项，如图 5-20 所示。在弹出的【移动图表】对话框中选择【新工作表】选项，单击【确定】按钮。

现在这个图表出现在一个新的工作表里，这可以从底部的工作表名选项卡看出，这个新工作表名为 Chart1(见图 5-21)。如果读者想给这个工作表取另外一个名字(如果工作簿包含很多工作表，最好给工作表取一个比较独特的名字)，右击这个选项卡，选择【重命名】命令。我们也可以方便地用鼠标单击和拖动来重新排列底部的选项卡。

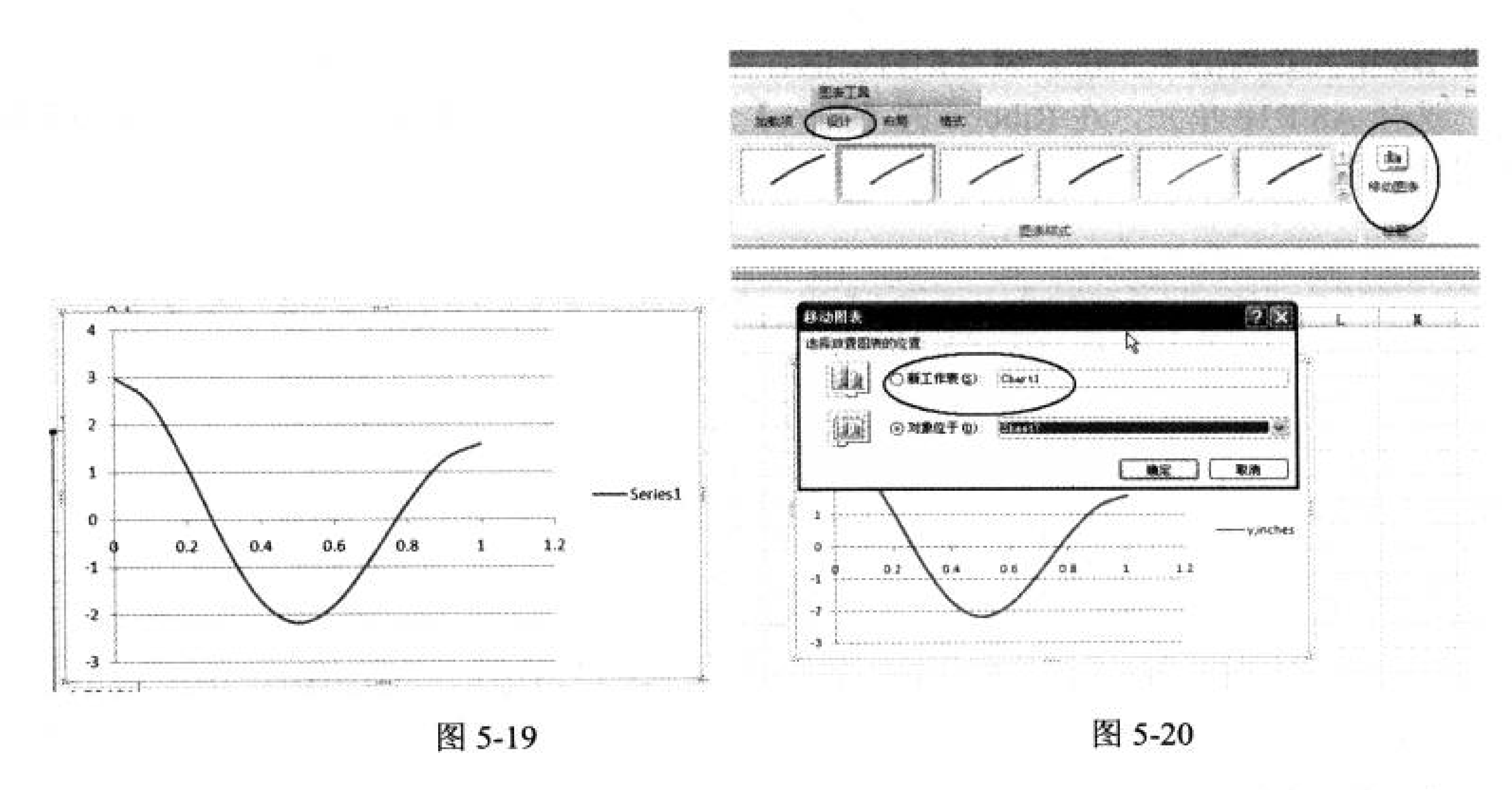


图 5-19　　　　图 5-20

任意选取 0~1 之间的某个时间。从这个图表可以看出，我们需要扩展它的时间域。

选取包含时间的最后两个值的单元(0.9s 和 1s 的单元)，如图 5-22 所示，单击并拖动填充柄，直到在光标的附近出现数值 5 为止，如图 5-23 所示。释放鼠标，我们发现，在时间这一列里，时间从 0s 开始以 0.1s 递增，递增到 5s。

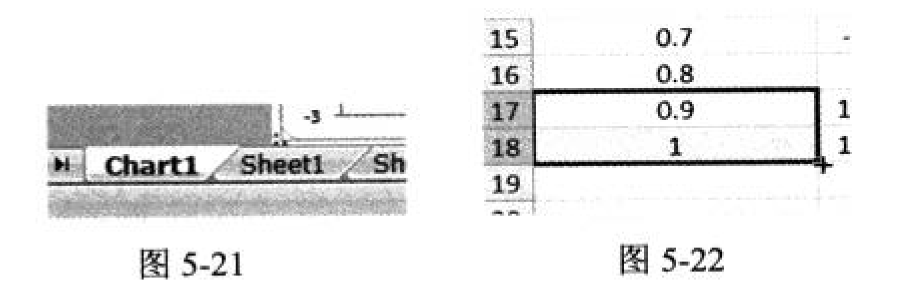


图 5-21　　　　图 5-22

正如前面曾提到，必须至少选择两个单元，才能够定义填充模式。

选择包含 y 值计算公式的最后一个单元(即时间为 1s 单元的邻接单元)，双击填充柄，如图 5-24 所示，Excel 把这个单元的公式复制到下面的单元里。

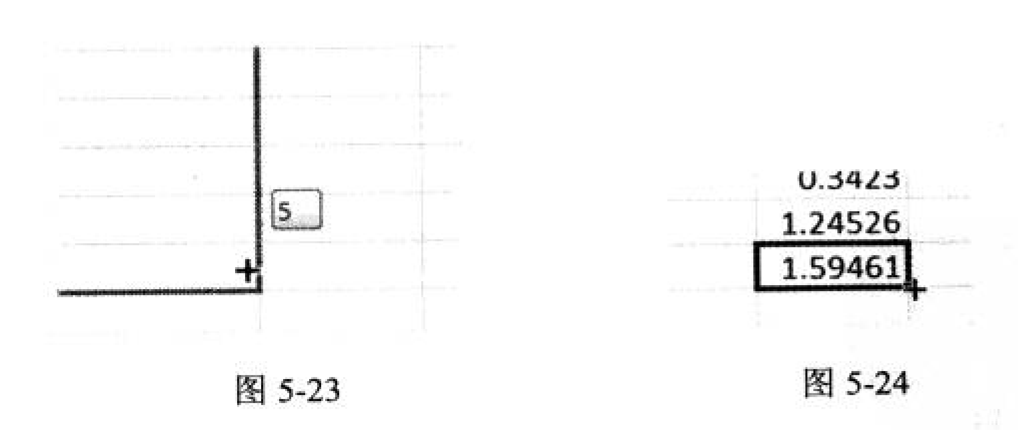


图 5-23　　　　图 5-24

回忆一下，采用这种复制方法，可以把公式向下复制直到它的邻接一列里没有输入值的单元为止。

现在我们把这些新数据添加到图表上。

右击图表区域，在弹出的快捷菜单中选择【选择数据】命令，如图 5-25 所示，注意选取图表的数据范围。切换到包含数据的工作表，选择新的数据范围。新的数据范围如图 5-26

所示。

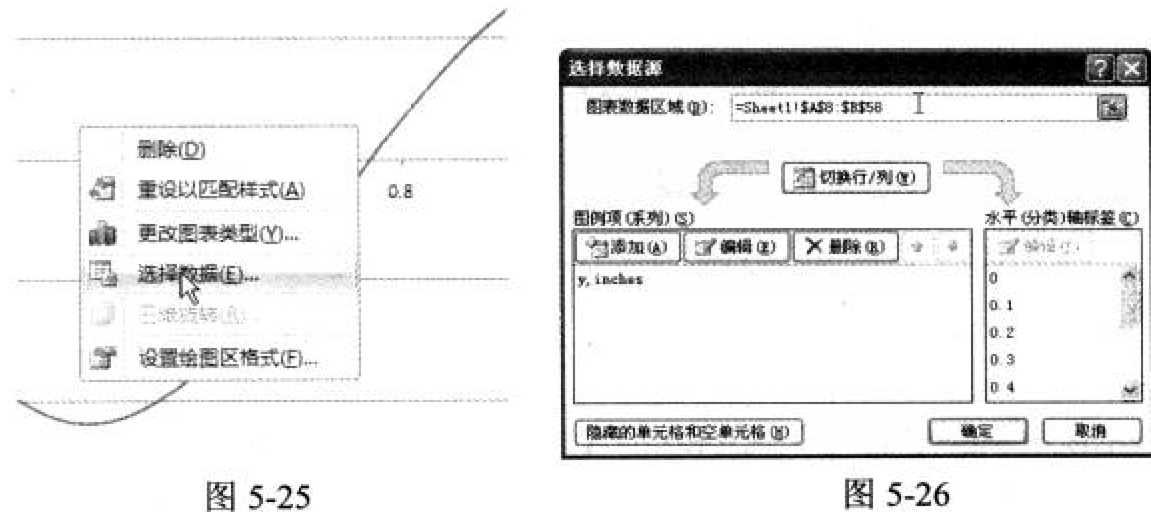


图 5-25　　　　图 5-26

现在的图表如图 5-27 所示。把时间域扩展到 5s，可以更好地看出该系统的振动过程，可以看出振幅随时间逐渐衰减的过程。

现在我们修改图形的外观。

先选择这个图表，切换到设计选项卡。在图表样式选择的右侧单击【其他】样式箭头，如图 5-28 所示，它包括线型、背景颜色和线宽等各种不同的图表样式。我们选择了一种普通的白色背景宽线条的样式。

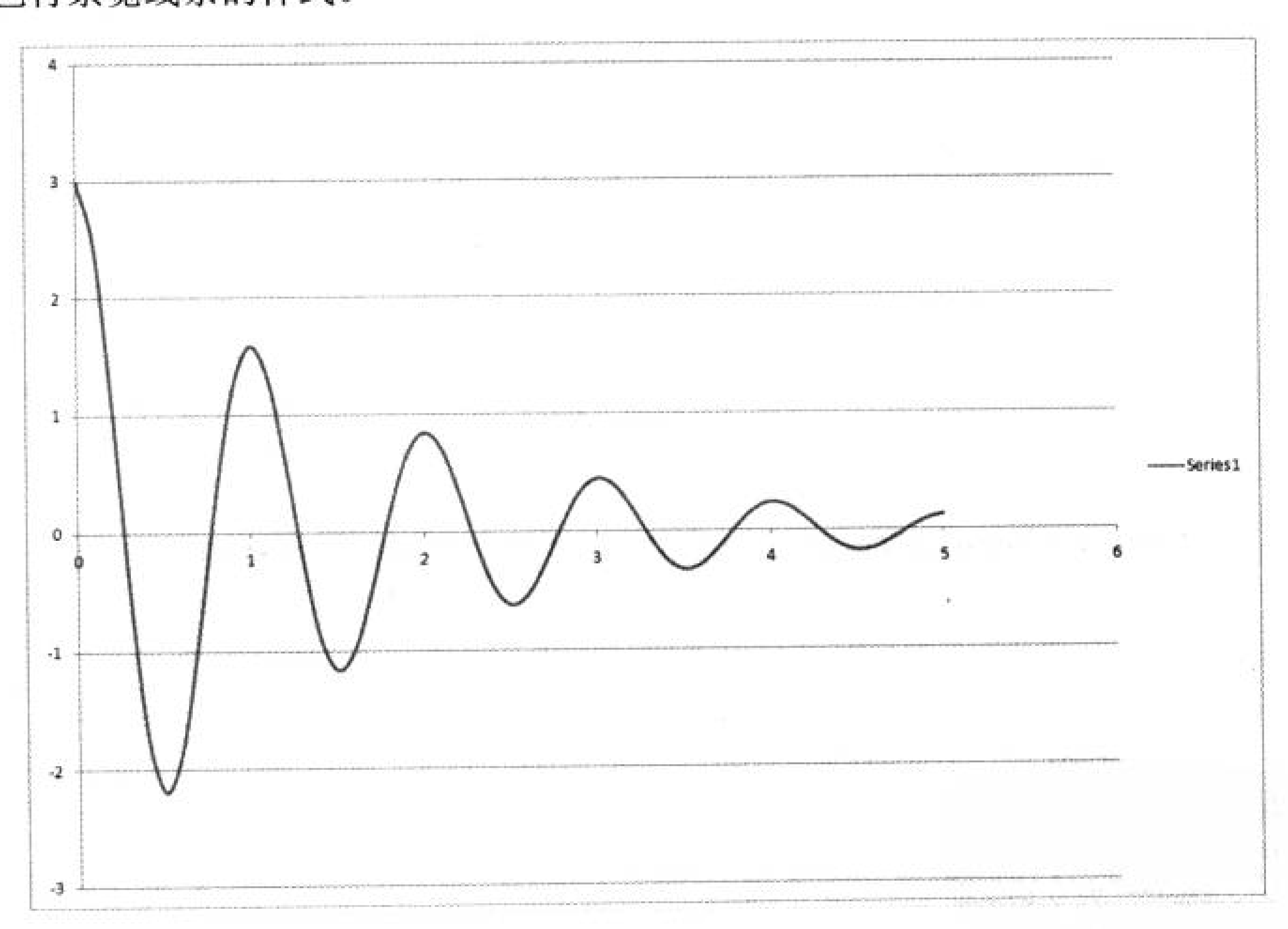


图 5-27

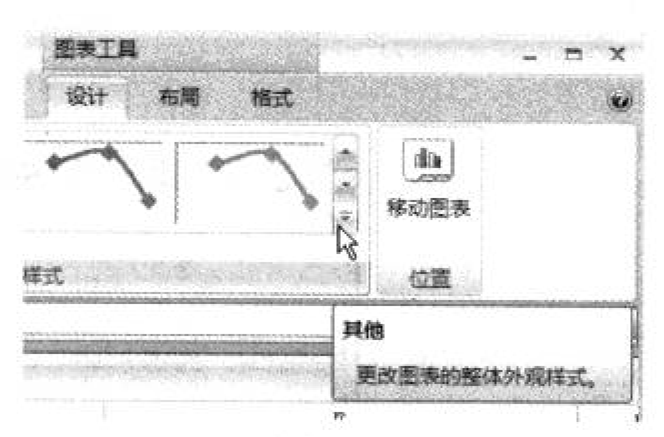


图 5-28

用来修改图形外观的工具位于 Ribbo 的【布局】选项卡里。

选择【布局】|【图表标题图表上方】命令，如图 5-29 所示，输入“阻尼弹簧振子系统的响应”标题，如图 5-30 所示，并按 Enter 键。

选择【布局】|【坐标轴标题】|【主要横坐标轴标题】|【坐标轴下方标题】命令并输入“Time:Seconds”作为横坐标轴标题。选择【布局】|【坐标轴标题】|【主要纵坐标轴标题】|【旋转的标题】命令并输入“Displacement:inches”作为纵坐标轴的标题。

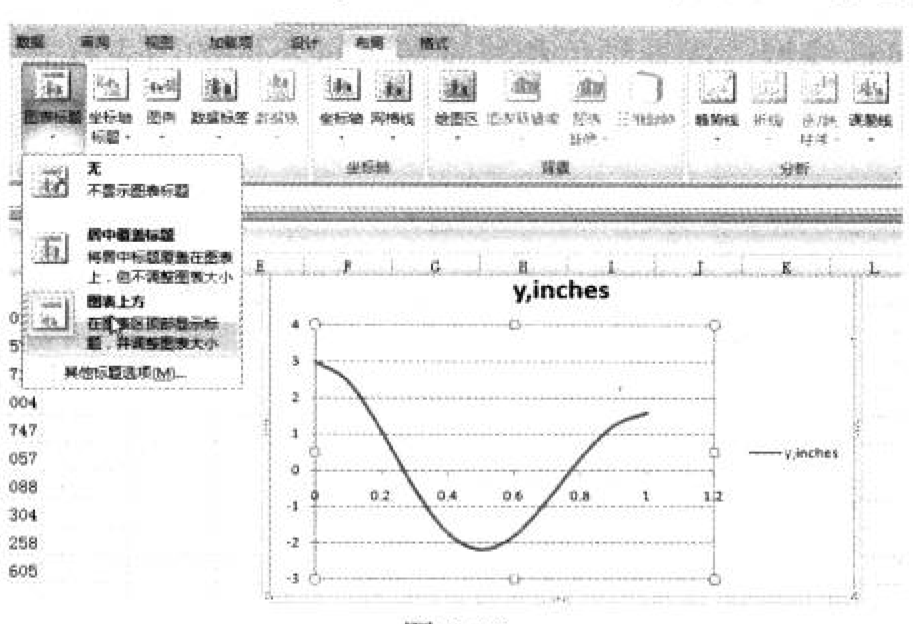


图 5-29

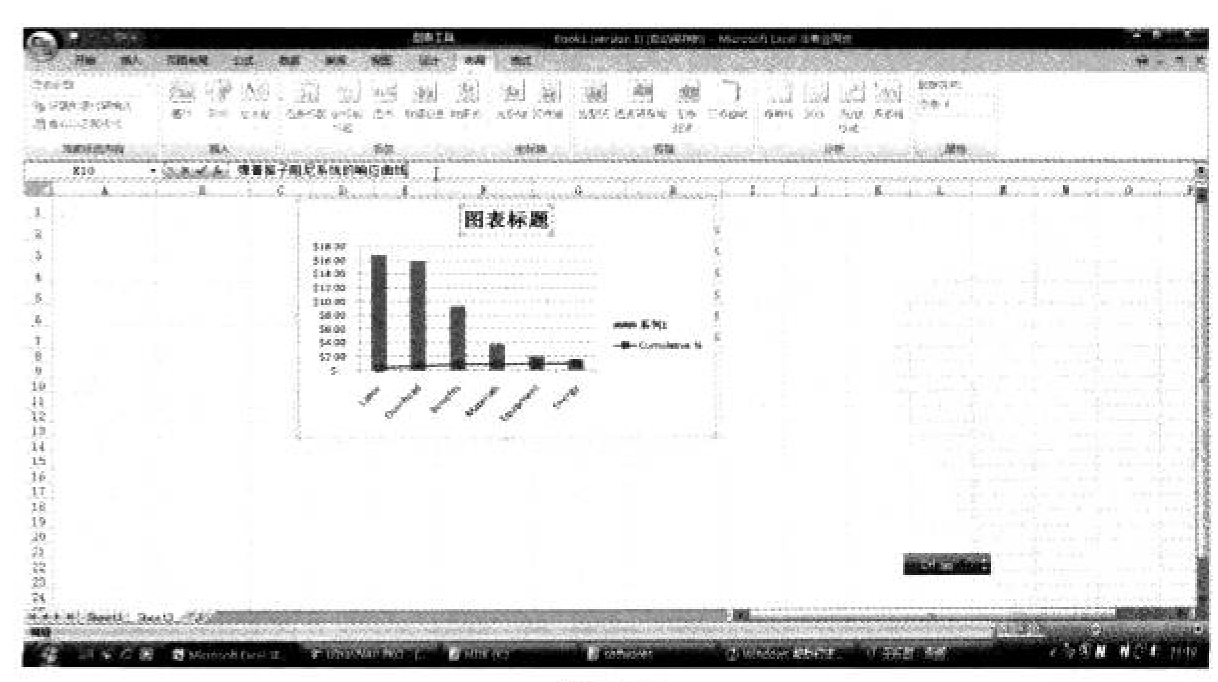


图 5-30

这个图里只有一个数据系列，因此不需要图例。

选择【布局】|【图例】|【无】命令，如图 5-31 所示。

默认时，Excel 显示水平网格线，不显示垂直网格线。但是为了更好地从图表上估计某个值，我们希望图表显示垂直网格线。

选择【布局】|【网格线】|【主要纵网格线】|【主要网格线】命令，如图 5-32 所示。

在这个图表里，x 轴的显示范围为 0~6s，由实际计算的时间范围是 0~5s，因此我们需要修改 x 轴的自动取值范围，把它改为 5s。

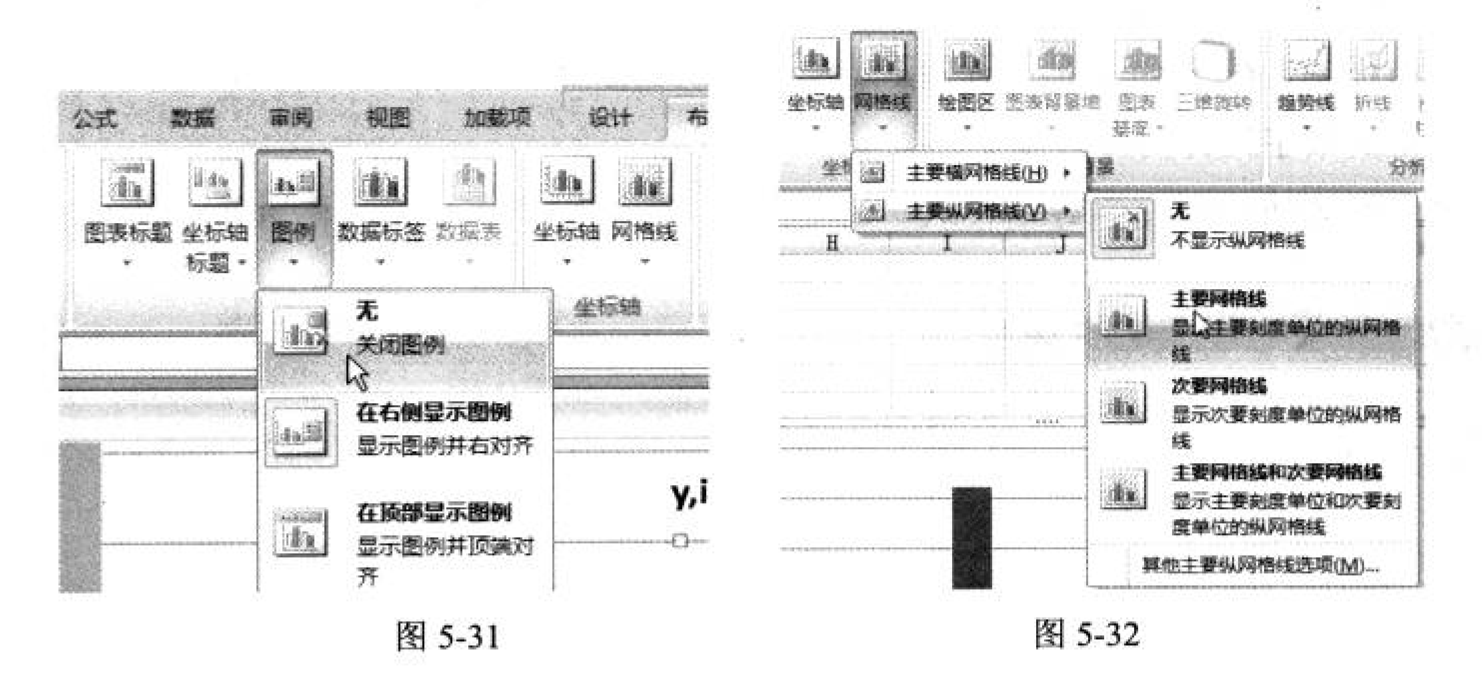


图 5-31　　　　图 5-32

选择【布局】|【坐标轴】|【主要横坐标轴】|【其他主要横坐标轴选项】命令，如图 5-33 所示。在弹出的【设置坐标轴格式】对话框中，选择【坐标轴最大值】的【固定】选项，输入 6.0 作为其最大值。此外，从【坐标轴标签】下拉列表中选择【低】选项作为横坐标标签的位置，如图 5-34 所示。最后单击【确定】按钮。

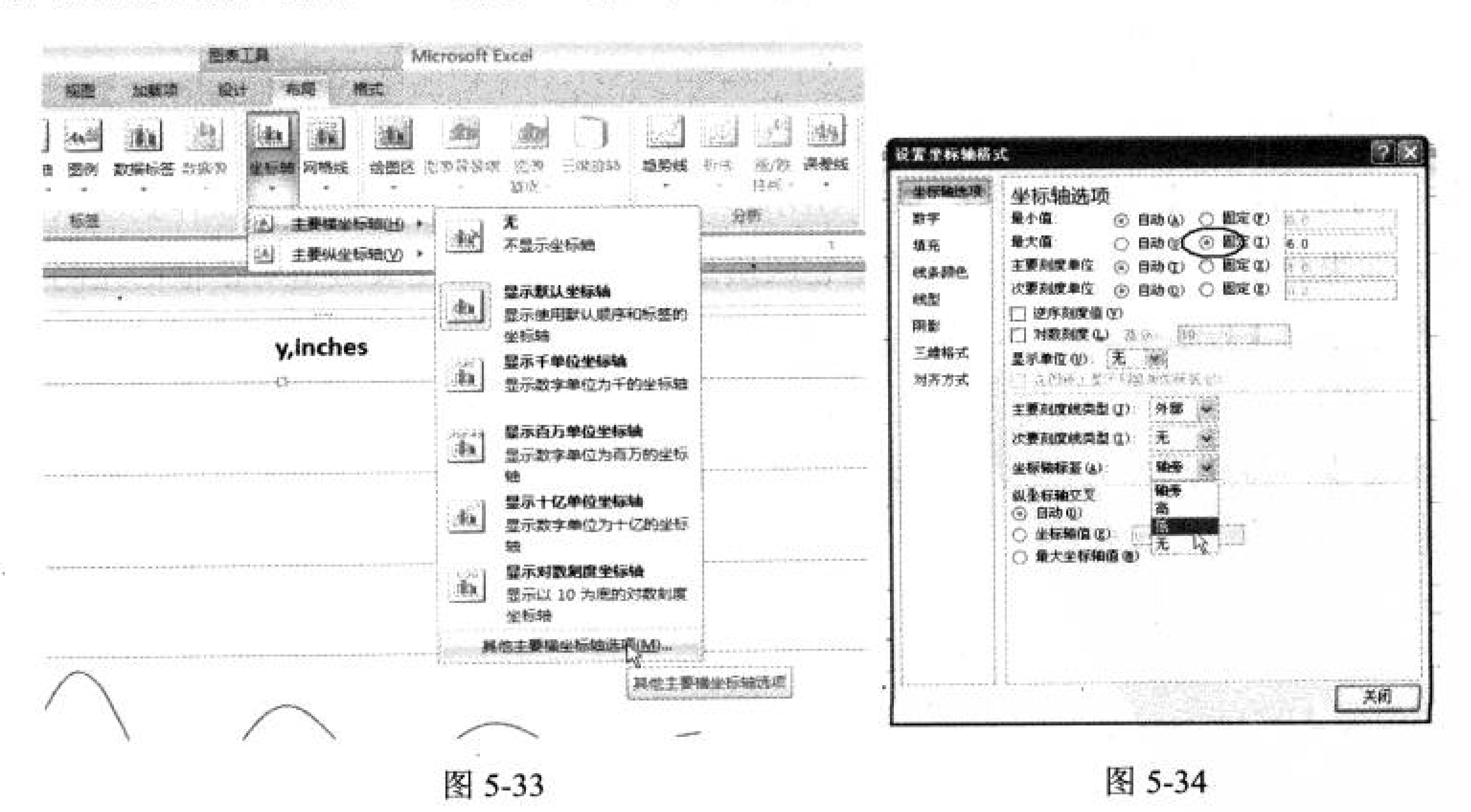


图 5-33　　　　图 5-34

需要注意，通过鼠标右击坐标轴，并从快捷菜单中选择【格式】命令，也可以访问这些选项。本节讨论的许多选项既可以通过 Ribbon 上的命令访问，也可以通过右击图表需要修改的数据对象访问。

右击纵坐标轴，选择【设置纵坐标轴格式】命令，把纵坐标轴的最大值设置为 3。

右击纵网格线，选择【设置主要网格格式】命令，把线条颜色设置为实线，从颜色下拉列表中把线条的颜色改为浅灰色，如图 5-35 所示。对横坐标网格重复这些步骤。

把网格线设置为浅颜色可以突出数据线。

单击标题，在 Ribbon 的【开始】选项卡中，选择合适的字体大小。对每个坐标轴标签和坐标上的数字进行同样的设置。

最后得到如图 5-36 所示的图表。

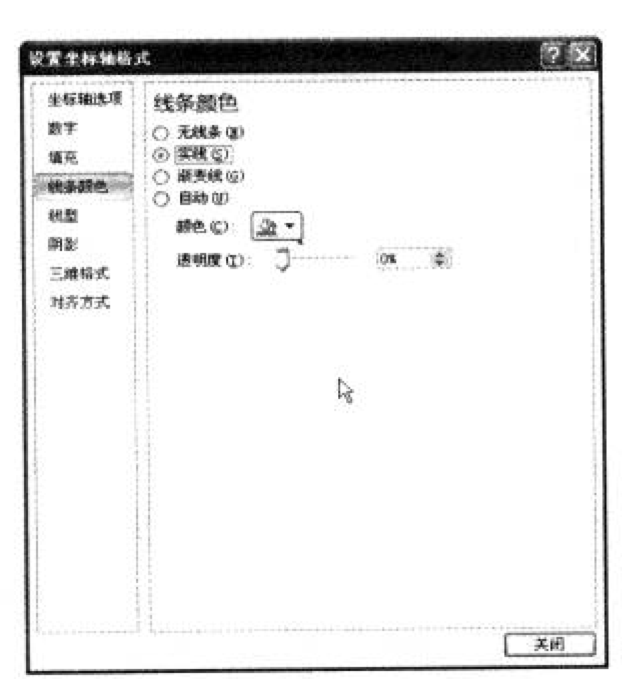


图 5-35

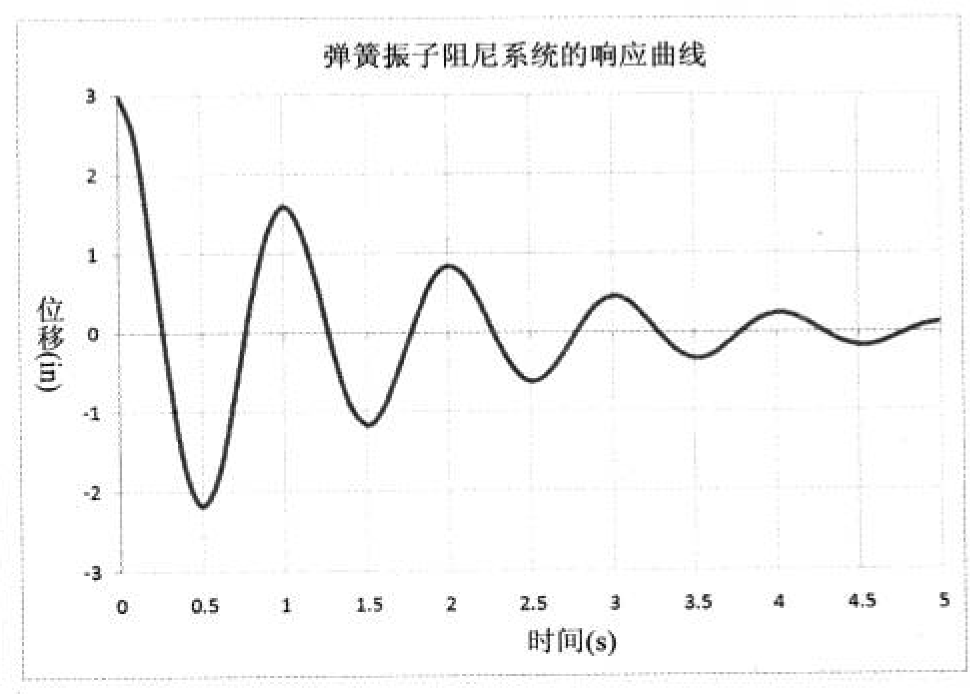


图 5-36

如果读者希望把这个图表的设置保存起来，用在以后的图表里，可从 Ribbon 中选择【设计】|【另存为模板】命令，如图 5-37 所示。给模板取一个名字，并把它保存到默认的目录里。如果我们建立了一个图表，希望它应用这些设置，可从图表类型对话框的左侧类型列表栏里选择【模板】文件夹，如图 5-38 所示。把光标移动到每个模板图标上，就会显示模板的名字。

下面我们将介绍如何在这个图里添加第二条曲线，用来显示系统对不同的阻尼系数的响应。

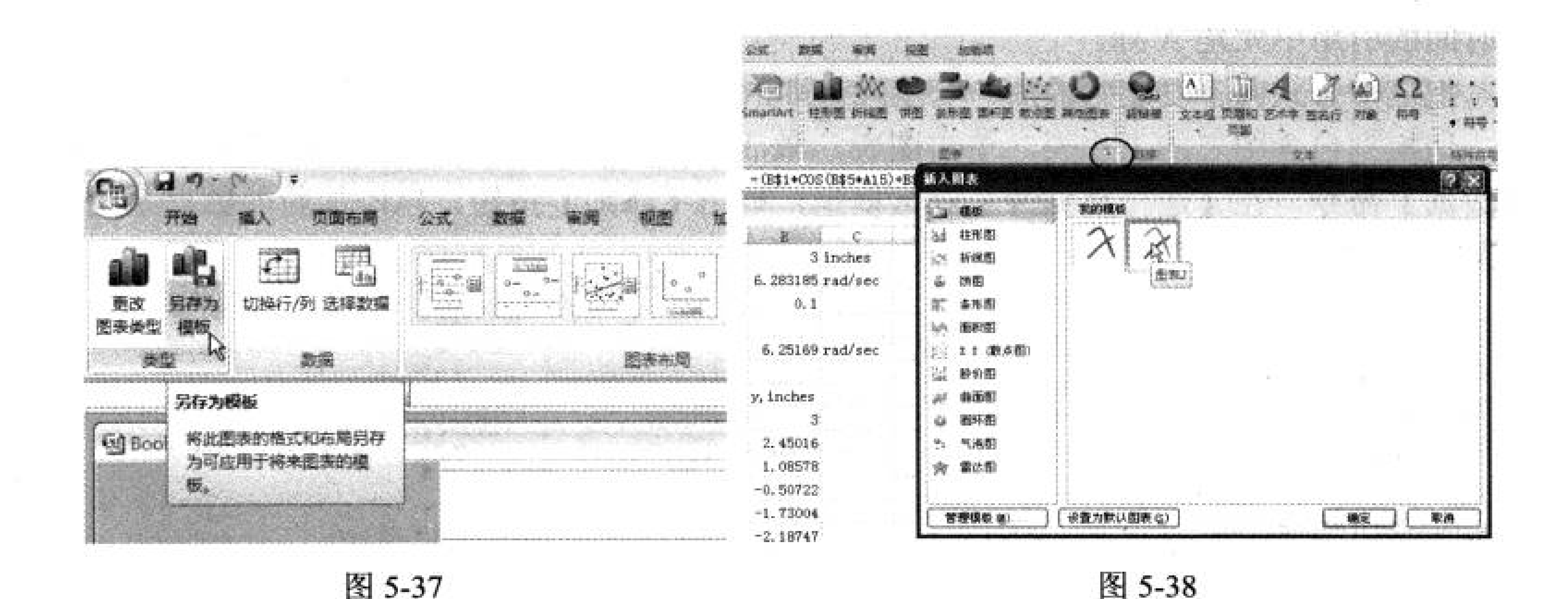


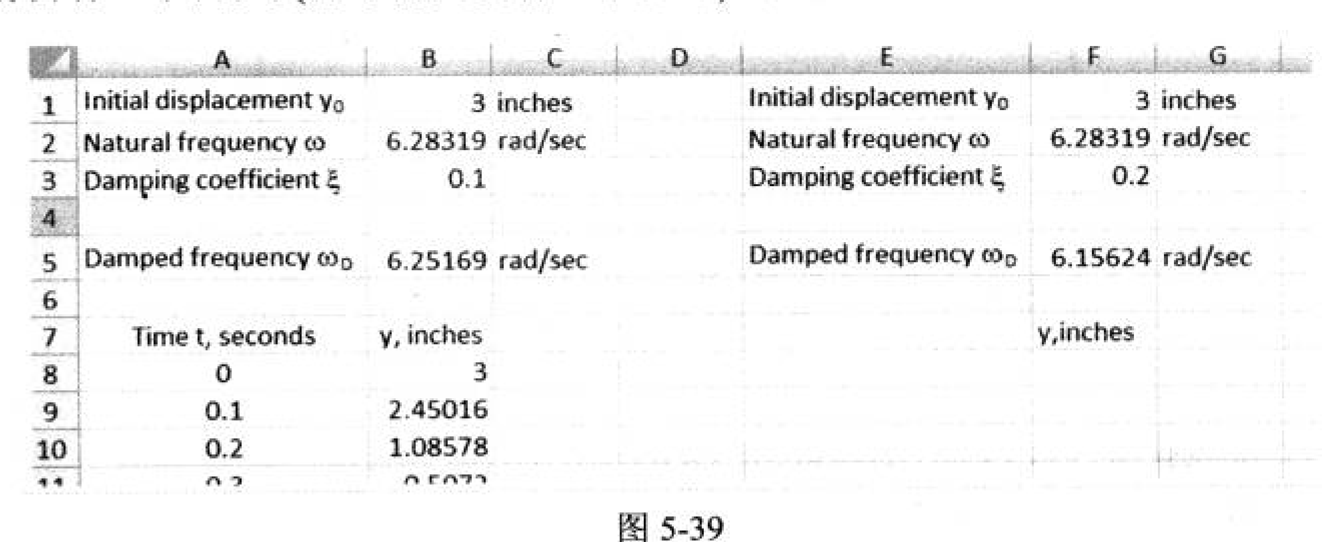
图 5-37　　　　　　　　　　　　　　　　　　图 5-38

例 5.2

给例 5.1 的图添加第二条曲线，该曲线反应系统在阻尼系数为 ξ=0.20 时的响应。

解：

复制并粘贴定义位移方程输入参数的单元的内容，如图 5-39 所示。把第二组数据里的阻尼系数改为 0.2。在复制位移公式之前，先双击包含 y 值计算的第一个单元(即图 5-39 的 B8 单元)，修改 y 的计算公式。对于在公式中引用的每个单元，按 3 次 F4 键，这样在每个列号之前会有一个$符号(行号前面没有这个符号)，如图 5-40 所示。



	A	B	C	D	E	F	G
1	Initial displacement y_0	3	inches		Initial displacement y_0	3	inches
2	Natural frequency ω	6.28319	rad/sec		Natural frequency ω	6.28319	rad/sec
3	Damping coefficient ξ	0.1			Damping coefficient ξ	0.2	
4							
5	Damped frequency ω_D	6.25169	rad/sec		Damped frequency ω_D	6.15624	rad/sec
6							
7	Time t, seconds	y, inches				y,inches	
8	0	3					
9	0.1	2.45016					
10	0.2	1.08578					

图 5-39

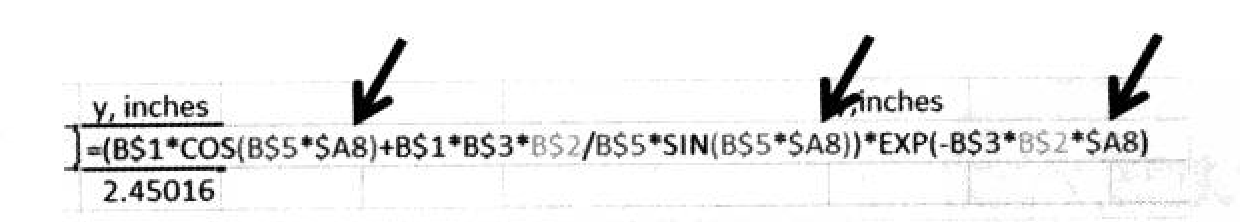


图 5-40

现在把这个公式复制到第二组 y 标题那一列的第一个单元(即图 5-40 的 F8 单元)，双击 F8 单元，显示它的公式如图 5-41 所示。

引用单元用不同的颜色表示，这个特征对我们检查和查找公式中出现的错误非常有用。注意，公式中引用的 3 个单元正好是该单元上方的初始位移、阻尼系数和阻尼频率，

而时间的引用单元是 A 列。

把这个公式向下复制到最后一个单元，即时间为 5s 对应的单元。

切换到包含图表的工作表。单击图表，激活 Ribbon 的图表工具，选择【设计】|【选择数据】命令，在【选择数据源】对话框里，单击【添加】按钮，以添加新的数据系列，如图 5-42 所示。

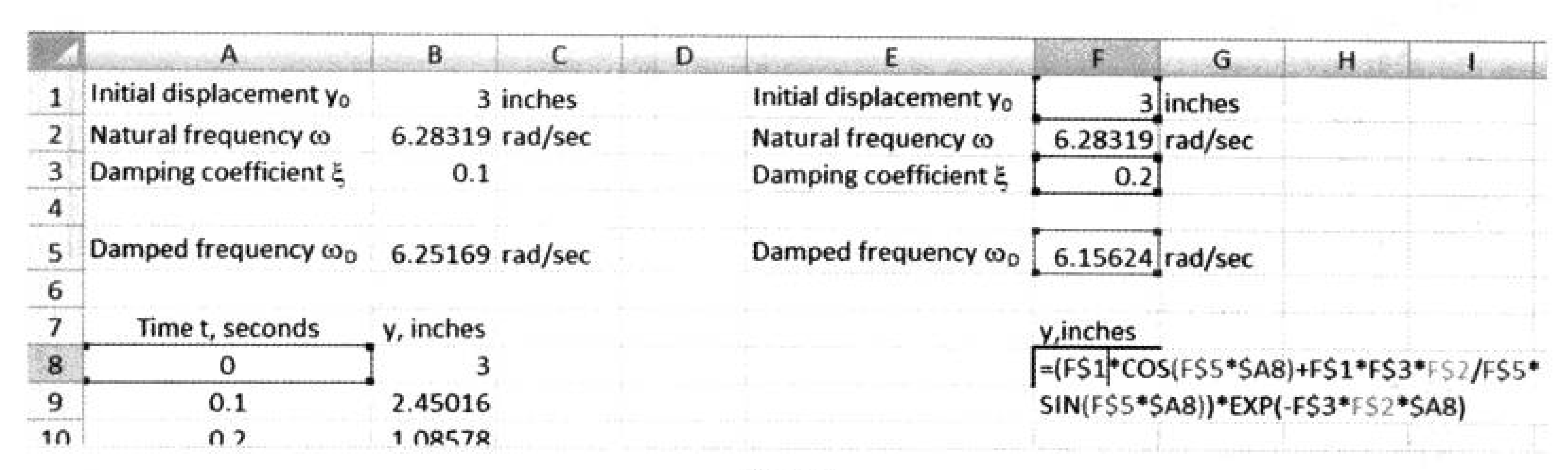


图 5-41

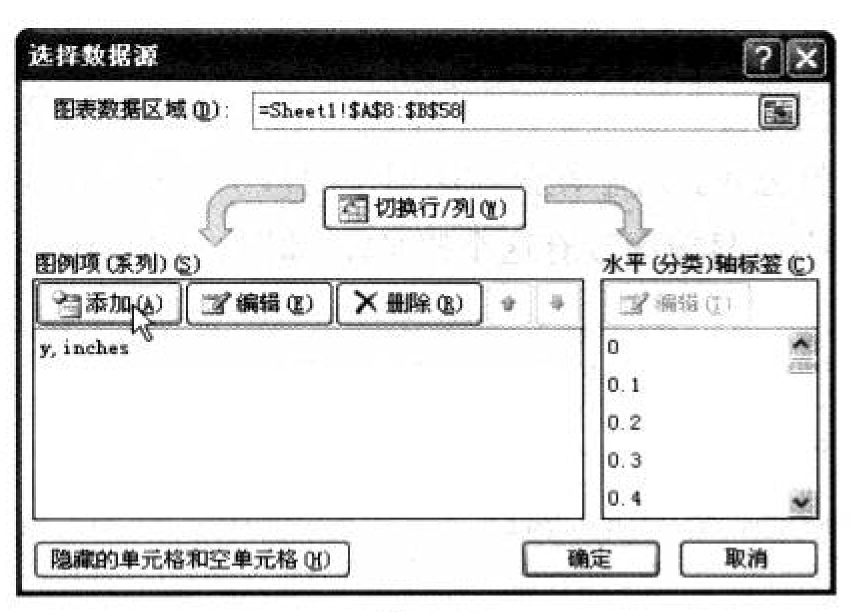


图 5-42

给这个新数据系列取名为“Damping factor=0.2”。单击标有“X 系列值”文本框旁边的图标，如图 5-43 所示。选择包含 0~5s 时间的单元，并按 Enter 键。单击“Y 系列值”右侧的图标，选择 F 列的 y 值数据。

新添加的数据系列如图 5-44 所示。

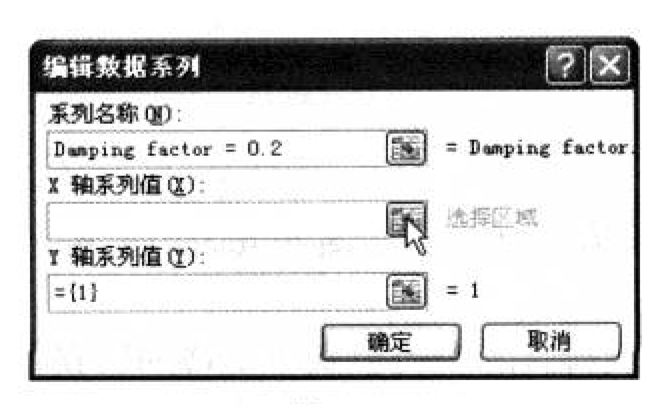


图 5-43

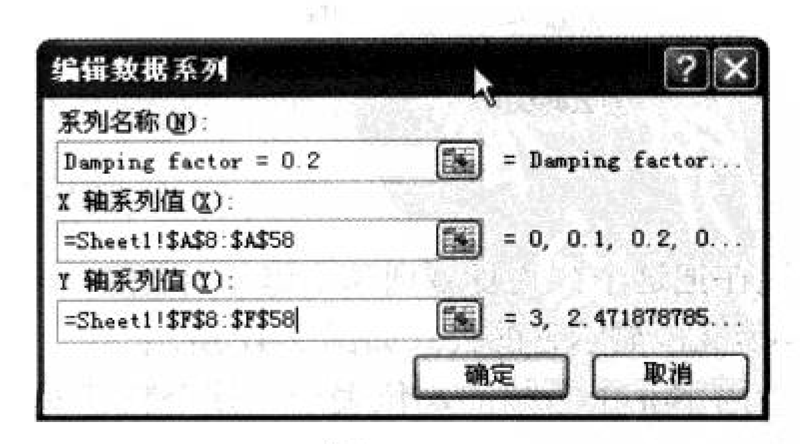


图 5-44

单击【确定】按钮，回到【选择数据源】对话框。选择【系列 1】，单击【编辑】按钮，把系列名称改为“Damping factor=0.1”，并按 Enter 键。单击【确定】按钮，关闭【数据系列选择】对话框。

右击图表上新添加的曲线，选择【设置数据系列格式】命令，如图 5-45 所示。在弹出的【设置数据系列格式】对话框中的【线条颜色】选项卡中，选择一种颜色。然后选择【线型】选项卡中，从列表中选择【短划线】，如图 5-46 所示。减小线的宽度，使得线段之间的间隔更加明显。

从 Ribbon 的布局选项卡上选择【图例】|【在右侧覆盖图例】命令，如图 5-47 所示。单击图例，从 Ribbon 的开始选项卡里增大字体大小。右击图例，从快捷菜单中选择【设置图例格式】命令，把填充效果改为【单色填充】并选择白色为填充色。把边框颜色改为实线并选择黑色作为边框色。

最后得到如图 5-48 所示的图表。

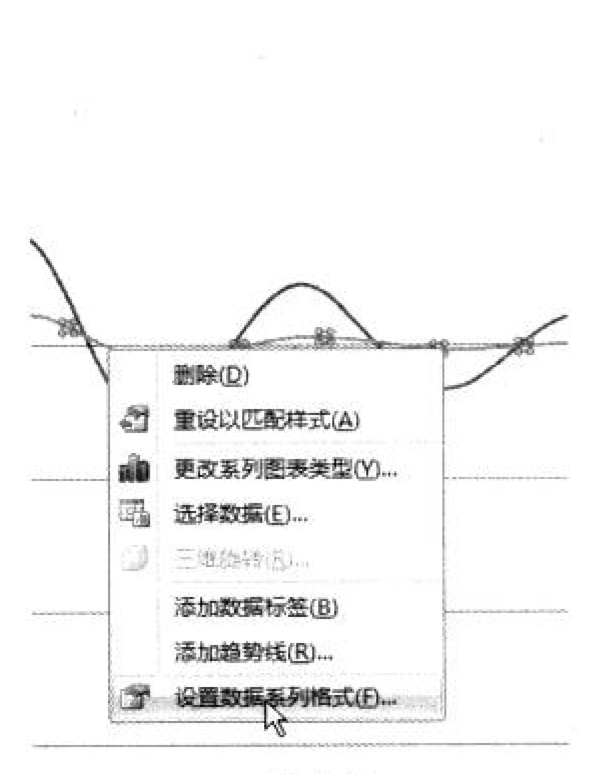


图 5-45

图 5-46



图 5-47

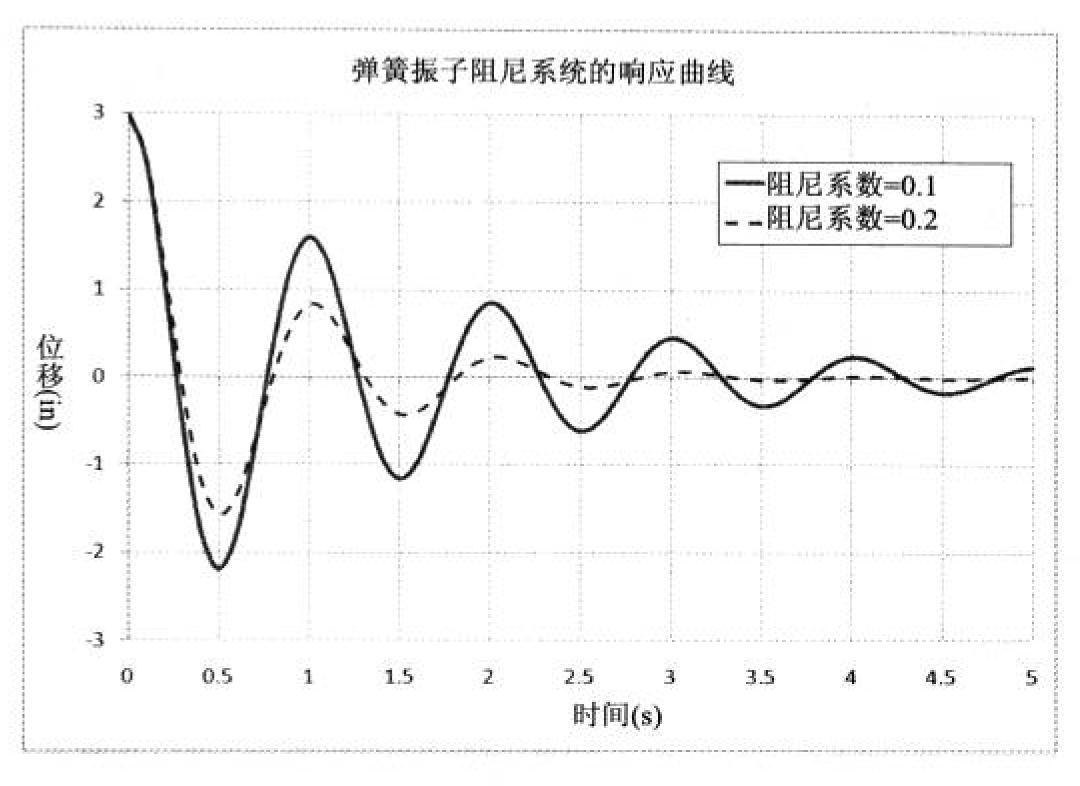


图 5-48

5.2.2 在 MATLAB 里绘制方程

为了说明 MATLAB 的图表绘制方法，我们重复例 5.1 的弹簧振子的阻尼振动问题。

例 5.3

绘制方程 5.1:

$$y=\left[y_0\cos\omega_D t+\frac{y_0\xi\omega}{\omega_D}\sin\omega_D t\right]e^{-\xi\omega t}$$

的曲线，y_0=3in， ω=2π rad/s， ξ=0.10。

解：

在这个教程里，我们将介绍 MATLAB 的两个绘图命令：plot 和 fplot。对于 plot 命令，把其中一个数组作为 x 轴，把另一个数组作为 y 轴，绘制它们的关系曲线。对于 fplot 命令，绘制一个函数在某个范围内的曲线。我们先介绍 plot 命令。

使用 plot 命令，必须先建立两个一维数组。这个命令的格式是:

plot(x,y)

命令中的 x 是一个数组，它对应横坐标 x 的值(本例它是时间)，y 是另一个数组，它的值对应于纵坐标 y 的值(在本例中，它是位移值)。这两个数组的个数必须相等，这样，对应每个时间总有一个位移值与之相对应。与前面的 Excel 例子一样，我们计算时间域为 0~5s 的位移曲线。时间步长为 0.05，因此每个数组要保存 101 个值 。用 for 循环建立计算这些数组的值。for 循环的计数器 i 的取值范围为 1~101。相对应的时间值是(i-1)*0.05，这样，第一个点的时间为 0。位移 y 值由方程 5-1 求解。由于使用 for 循环，因此取较小的时间步长并不会花费太多我们太多的精力，这与 Excel 不同。

打开一个新 m-file 文件，在编辑器里输入以下代码，把这个文件保存为 SMD。

```
1  % SMD.m: Plots displacement of under damped
2  % spring-mass-damper system
3  %
4  % y0 = initial displacement, inches
5  % dc = damping coefficient
6  % fr = natural frequency, radians/second
7  % fd = damped frequency, radians/second
8  % t = time, seconds
9  % y = displacement, inches
10 %
11 y0 = 3.0;
12 dc = 0.10;
13 fr = 2*pi;
14 fd = fr*sqrt(1-dc^2);
15 for i = 1:101
16     t(i) = (i-1)*0.05;
17     c = cos(fd*t(i));
18     s = sin(fd*t(i));
19     e = exp(-dc*fr*t(i));
20     y(i) = (y0*c + y0*dc*fr/fd*s)*e;
21 end
22 plot(t,y)
```

注意，我们本来完全可以把 for 循环里求 y 值的表达式写成一行的式子，如下所示：

```
y(i) = (3.0*cos(2*pi*sqrt(1-0.10^2)*t(i))+3.0*0.10*
2*pi/(2*pi*sqrt(1-0.10^2))*sin(2*pi*sqrt(1-0.10^2)*
t(i)))*exp(-0.10*2*pi*t(i))
```

但是有许多理由要求把一个长表达式分解为几项进行计算：

- 如果用变量表示方程中的常量，包括初始位移、阻尼系数和固有频率，如第 11 行~第 13 行，这样以后修改它们的值就非常方便。我们不需要在脚本中查找这些常量出现的地方并逐一进行修改，只要修改变量赋值语句里的值。
- 阻尼系数在位移方程中出现 3 次，如果 3 次输入阻尼值的计算公式，则出错的可能性就比较大(有可能 3 次输入都不一样)。像第 14 行这样处理——计算一次并把它赋给一个变量——的另一个理由是，它不是时间的函数。因此在进行 for 循环之前求它的值，比起在 for 循环内部计算它的值，效率更高。
- 像第 17 行~第 19 行那样，先在单独一行分别计算正弦项、余弦项和指数项，使得第 20 行的最终表达式简单许多。现在这个式子只有一对括号，而上面的式子有 11 对括号。括号匹配不正确是一个常见的错误。

  更容易检查运算结果。复杂方程的计算经常需要手工进行校对，把一个复杂的方程分解成简单的几个式子，会使检查过程容易许多。

在命令窗口执行 SMD 脚本程序

出现一个新窗口，窗口里显示位移曲线，窗口的标题为 Figure 1，如图 5-49 所示。在

这个窗口里可以修改这个曲线的格式。

要修改这个曲线图，必须先单击工具栏上的箭头图标，如图 5-50 所示。

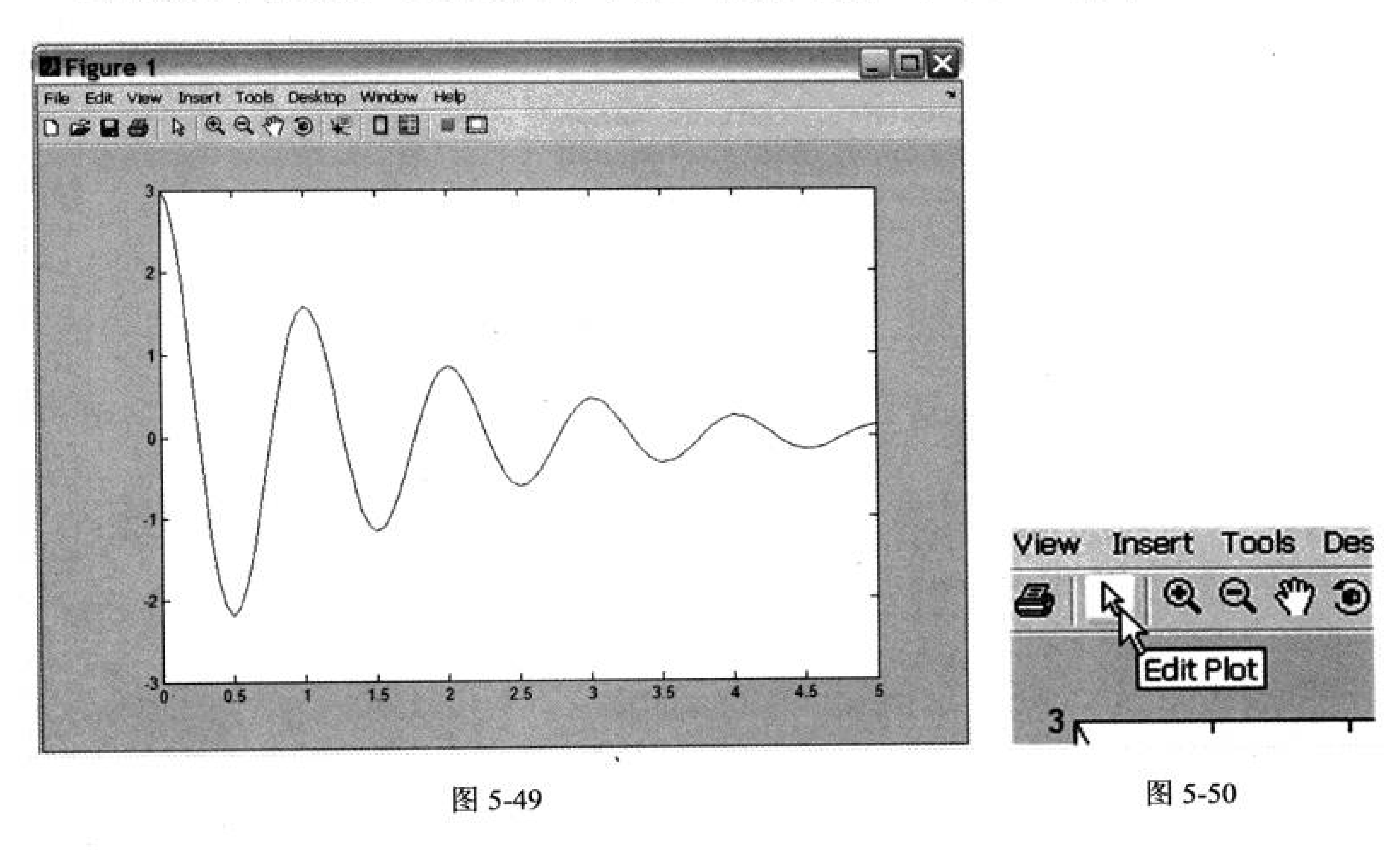


图 5-49　　　　图 5-50

右击该曲线，选择 Line Width(线宽)命令，如图 5-51 所示。选择较大的线宽值。用同样的方法，修改曲线的颜色。右击图形中的空白区，在弹出的快捷菜单中选择 Grid(网格)命令，图中就出现了网格线，如图 5-52 所示。

利用属性编辑器(Property Editor)，可以修改曲线的许多属性。

右击屏幕上的空白区，在弹出的快捷菜单中选择 Show Property Editor(显示属性编辑器)命令，如图 5-53 所示。

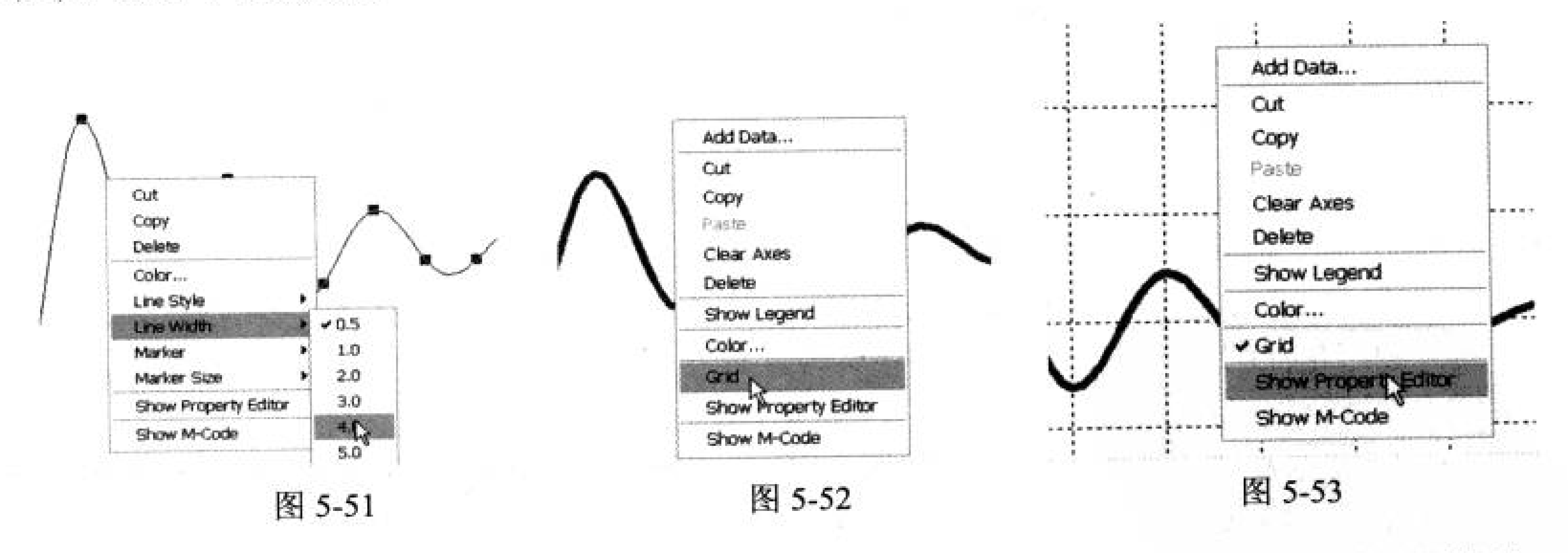


图 5-51　　　　图 5-52　　　　图 5-53

在图的正下方出现一个属性编辑器，如图 5-54 所示。在属性编辑器里，我们可以添加标题和设置标题的格式、修改线型和颜色、显示/隐藏网格线。注意，还有一个选项，允许我们修改 x 轴。因为 MATLAB 可以建立三维图。本书只介绍二维图。如果读者想学习如何绘制三维图，可以参考帮助文档。

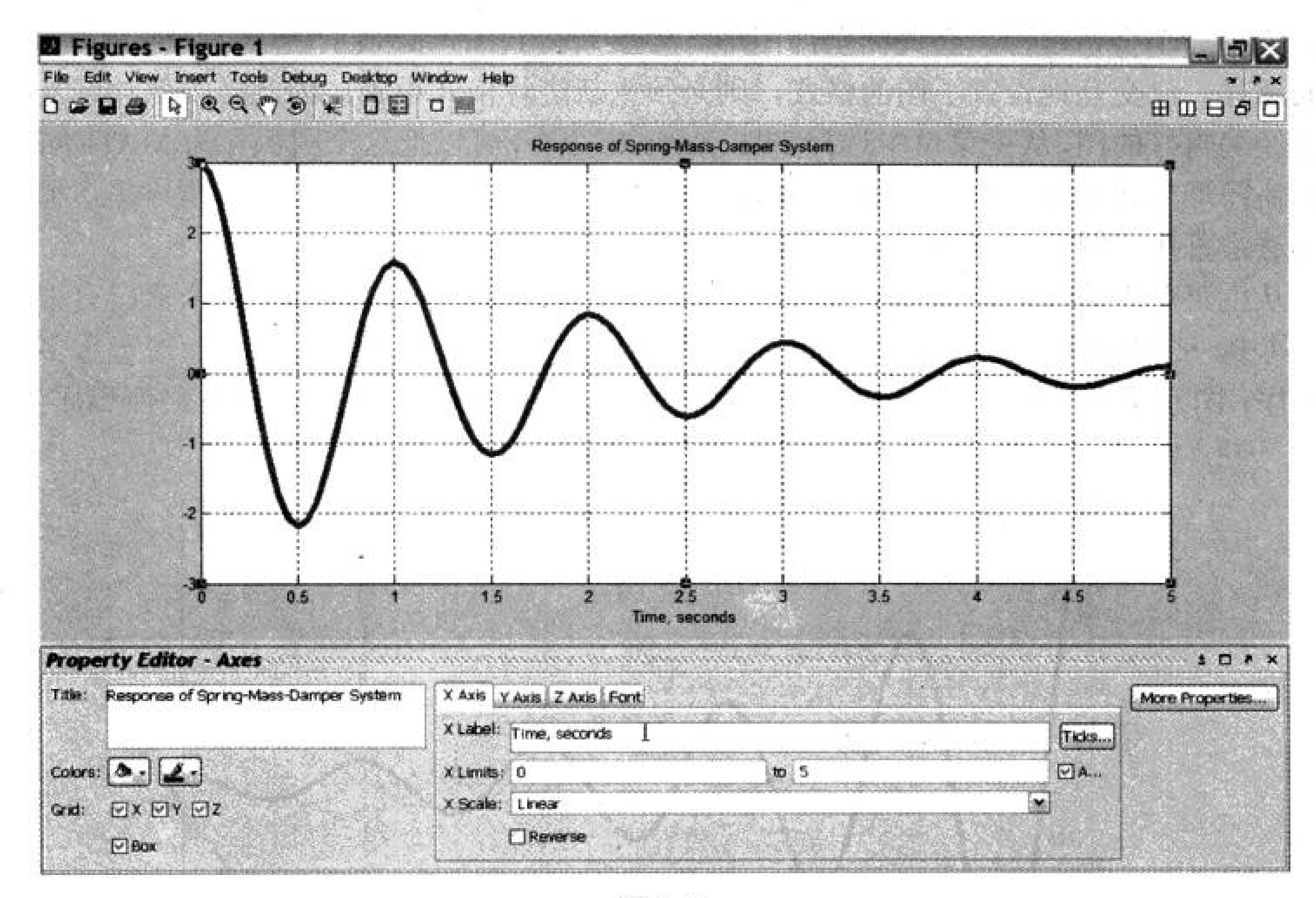


图 5-54

给这个图和坐标轴添加标题。单击每个文本对象，根据需要改变字体和大小。完成后，单击属性编辑器右上角的关闭按钮，如图 5-55 所示。要把绘制图区域之外设置为白色，可右击这个区域，选择 Color(颜色)命令，再选择白色。

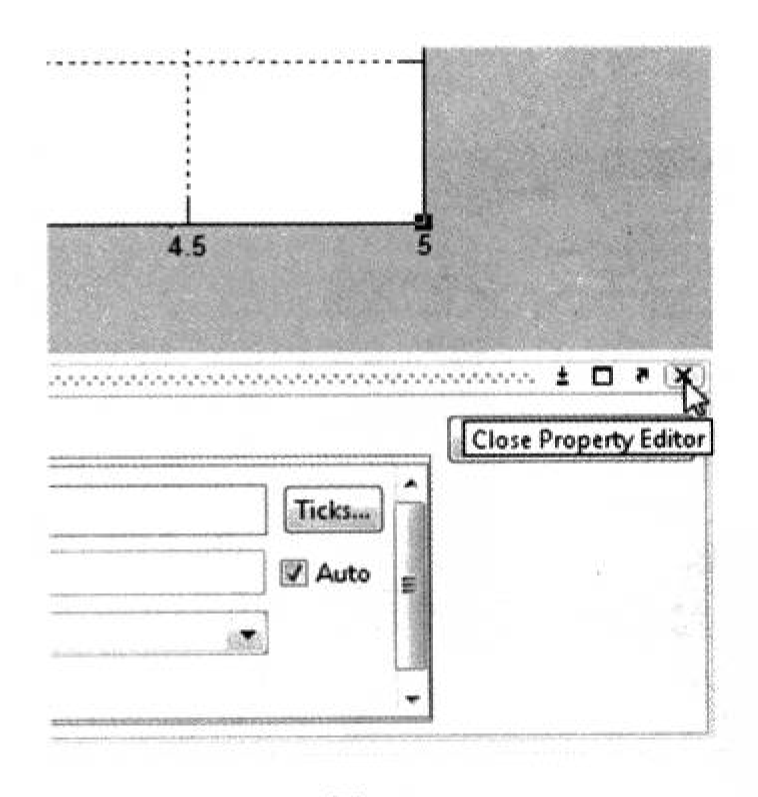


图 5-55

最后得到如图 5-56 所示的曲线。

在 Excel 里，当我们保存 Excel 文件时，图和它们的格式也将被一起保存。在 MATLAB

里，关闭一个会话过程，会关闭全部图形。要重新得到图，必须重新运行 m-file。如果像前面那样，已交互地设置了图的格式，则必须重新进行格式设置。因此必须保存或复制一个已经编辑好的图。从主菜单中选择 File | Save as 命令，可以把这个图保存为 MATLAB 图，或其他图形格式文件，如 jpeg 或 bitmap。这些文件可以被导入到其他应用程序里。另外一种办法是直接把图复制到 Word 或 PowerPoint 文档里。

从主菜单中选择 Edit | Copy 选项，出现一个如图 5-57 所示的窗口。这个窗口有很多选项，其中一个选项是把图复制为一个图元文件(metafile)或位图文件(bitmap)。对于大多数应用程序，图元文件可以得到一个分辨率很高的图像，而文件的大小不像位图文件那样很大。

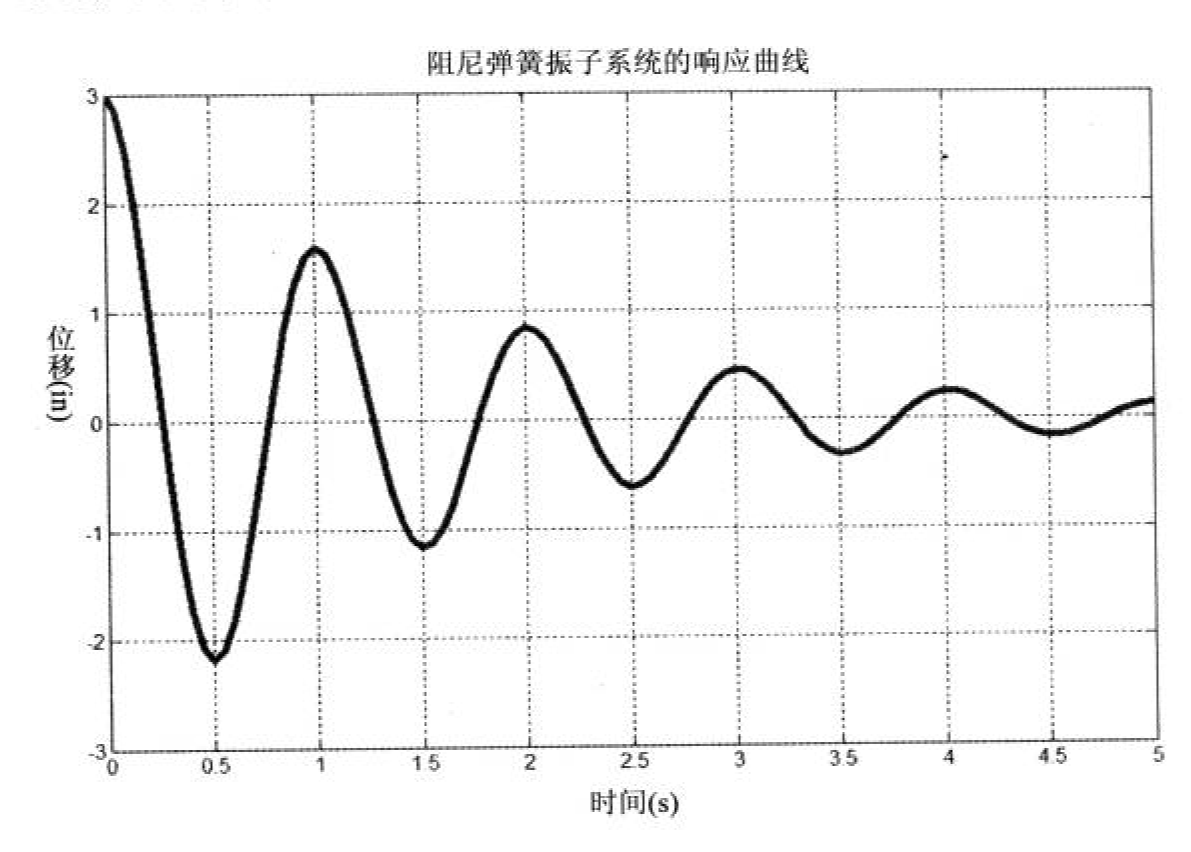


图 5-56

图 5-57

设置好选项后，从主菜单中选择 Edit | Copy Figure 命令，这个命令把图复制到 Windows 的剪切板里。用 paste(粘贴)命令，可以把它插入到其他文档里。

前面设置的几个格式选项也会被写入到 m-file 文件里。

打开 m-file SMD。修改 plot 语句，如下所示，说明线宽和颜色。再增加几行代码，打

开网格线，添加标题和坐标轴标签并对它们进行格式设置。注意，最后3个命令(title,xlabel 和 ylabel)都占用了两行内容。一行后面的省略号表示这行的命令继续到下一行。读者也可以把这些命令写在一行里。

```
21 end
22 plot(t,y,'LineWidth',4, 'Color', 'Red')
23 grid('on')
24 title('Response of Spring-Mass-Damper System',...
25 'FontSize',20, 'FontName', 'Arial')
26 xlabel('Time, seconds', 'FontSize',14,...
27 'FontName', 'Arial');
28 ylabel('Displacement, inches', 'FontSize',14,...
29 'FontName', 'Arial');
```

保存这个文件，通过命令提示符运行 SMD 脚本程序。得到一个新图形，如图 5-58 所示。

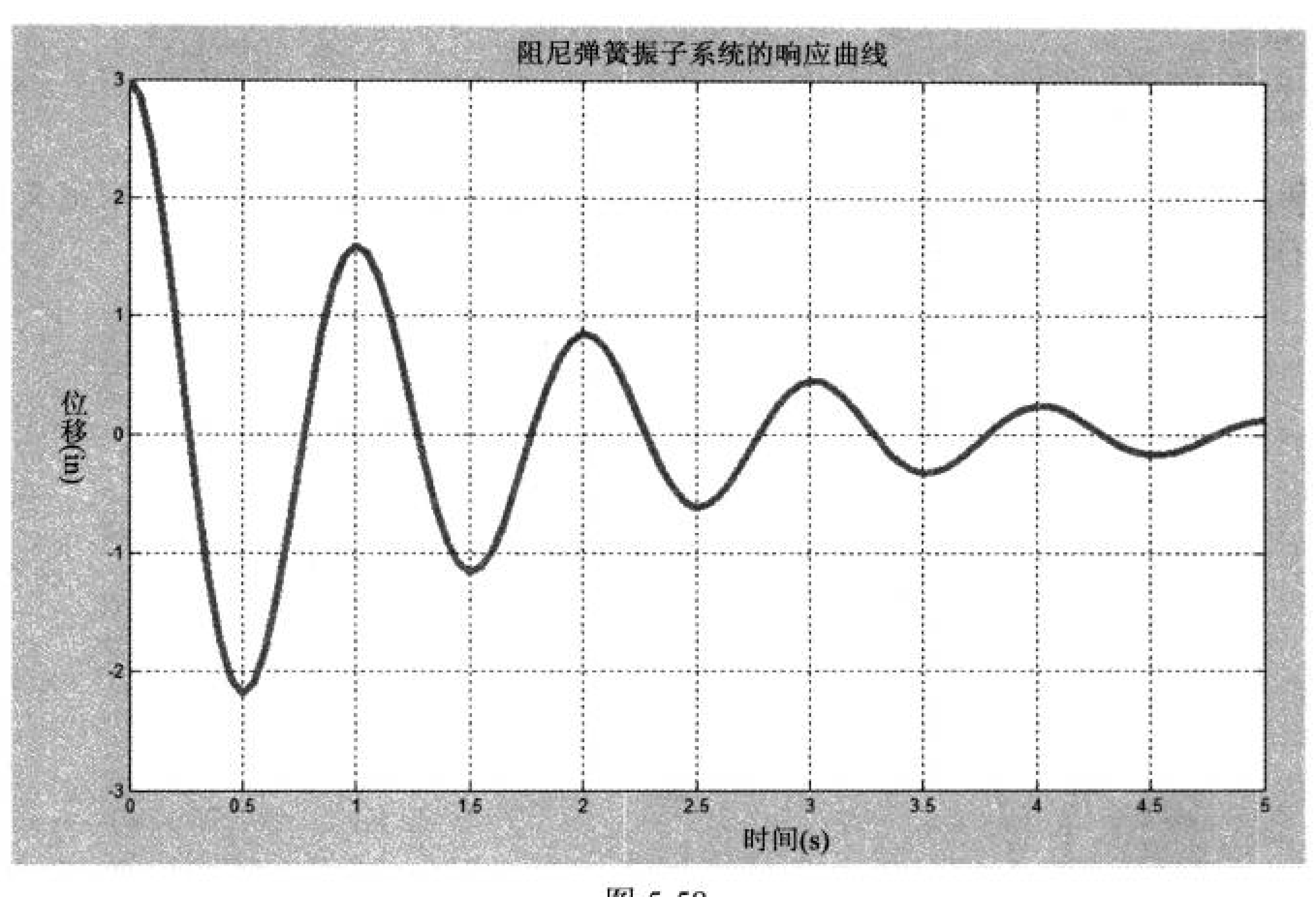


图 5-58

并非所有的格式选项都有对应的 MATLAB 命令，因此读者可能希望进一步用交互方式设置图形的格式。然而在 m-file 里添加一些图形格式命令可以节省我们的时间。因为我们可以多次重复使用这个 m-file。

在例 5.2 里，我们在图里添加了第二条曲线，它反应了阻尼系数的变化对系统响应的影响。现在我们用 MATLAB 重复此例子。

例 5.4

在例 5.3 中生成的图里添加第二条曲线，它对应于阻尼系数 $\xi=0.2$ 的情形。

解：

在 SMD 文件的末尾，添加以下几行代码。这些代码定义了阻尼系数的另一个值，计

算新的阻尼频率以及每个时间的位移值。

```
30 %
31 % Compute and plot displacements for a
32 % different damping coefficient
33 %
34 % dc2 = new damping coefficient
35 % fd2 = new damped frequency
36 % y2 = new displacement array
37 %
38 dc2 = 0.20;
39 fd2 = fr*sqrt(1-dc2^2);
40 for i = 1:101
41     c = cos(fd2*t(i));
42     s = sin(fd2*t(i));
43     e = exp(-dc2*fr*t(i));
44     y2(i) = (y0*c + y0*dc2*fr/fd2*s)*e;
45 end
46 plot(t,y2, 'LineWidth',3, 'LineStyle',...
47 '-', 'Color', 'Blue');
```

把新的脚本程序保存为 SMD，并运行这个文件。注意，现在没有必要再来定义时间值，因为我们可以使用前面已定义的时间数组。新得到的图形如图 5-59 所示。

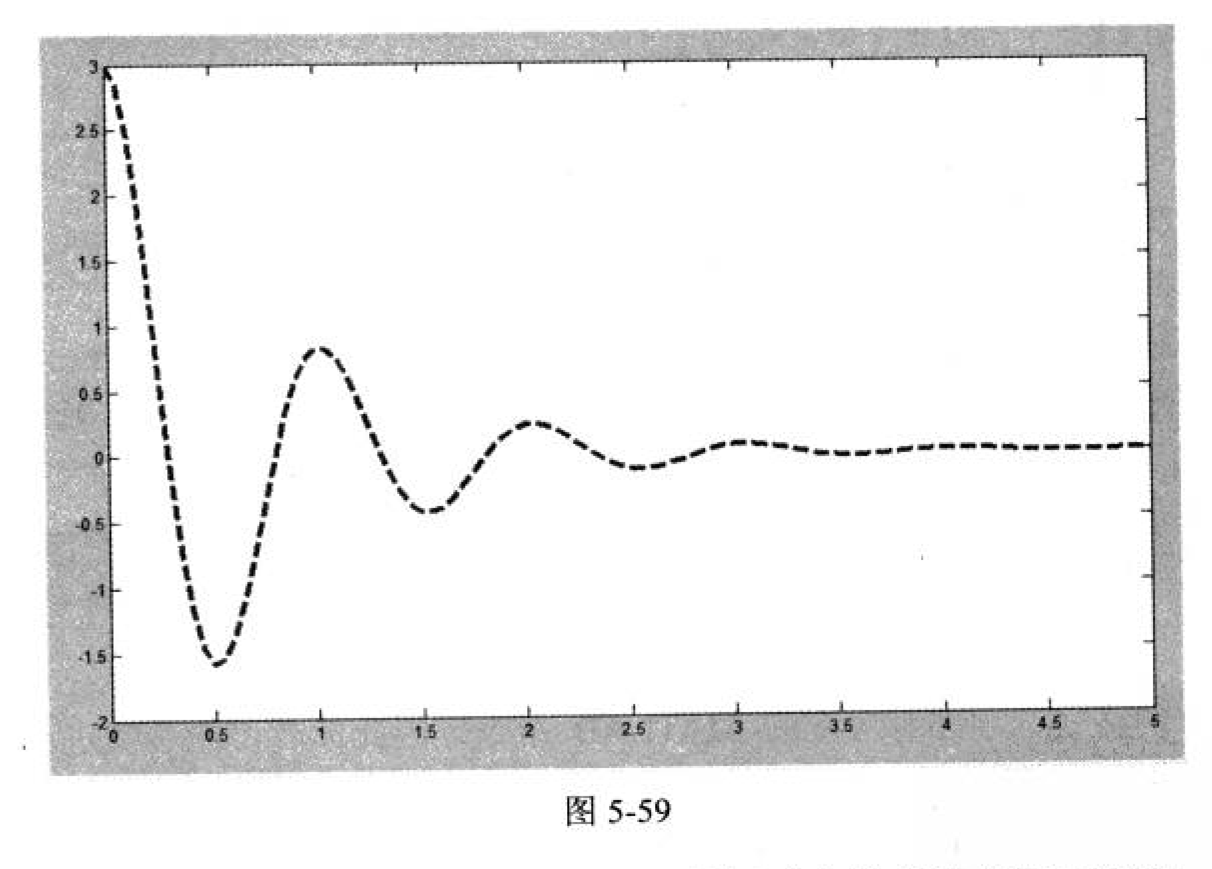


图 5-59

注意，原来的数据曲线，即阻尼系数为 0.1 的振动曲线并没有显示出来。这是因为，当 MATLAB 执行 plot 时，会删除此前绘制的图形。因此，第一个绘制的图的标题是 Figure 1，但是后来的 plot 命令也建立了一个新的 Figure 1。如果希望得到多个图形窗口，则要用 figure 命令打开一个新的图形窗口。

在 SMD 脚本程序里，在第二个 plot 命令之前，插入 figure 命令，如下所示：

```
45   end
```

```
46  figure
47  plot(t,y2, 'LineWidth',3, 'LineStyle',...
48  '-', 'Color', 'Blue');
```

保存并运行这个 m-file。

现在看到两个图形窗口，标题分别为 Figure 1 和 Figure 2。虽然，这个例子说明了如何在 m-file 里建立多个图形的方法，但有时我们希望在同一个图里显示两个数据集。为此，要使用 hold on 命令。hold on 使当前的图形处于打开状态，因此后续的绘制命令可以在这个图形窗口里添加图形。单独使用 hold 命令，表示在 hold on 与 hold off 之间切换。为了避免引起混淆，我们建议读者明确使用 hold on 和 hold off。

现在图上有多条曲线，因此需要用图例来区分它们。

打开 SMD 文件，把 figure 命令改为 hold on，再插入 legend 命令，如下所示：

```
45   end
46   hold on
47   plot(t,y2, 'LineWidth',3, 'LineStyle',...
48   '- ', 'Color', 'Blue');
49   legend('Damping Coefficient = 0.1',...
50   'Damping Coefficient = 0.2');
```

注意，在 Legend 命令里，两条曲线是由它们的建立次序确定的。

保存并执行这个 m-file，根据需要用属性编辑器对图进行格式设置(如图例的字体大小和坐标轴的数字大小)。把图例拖动到合适位置。

最后得到的图如图 5-60 所示。

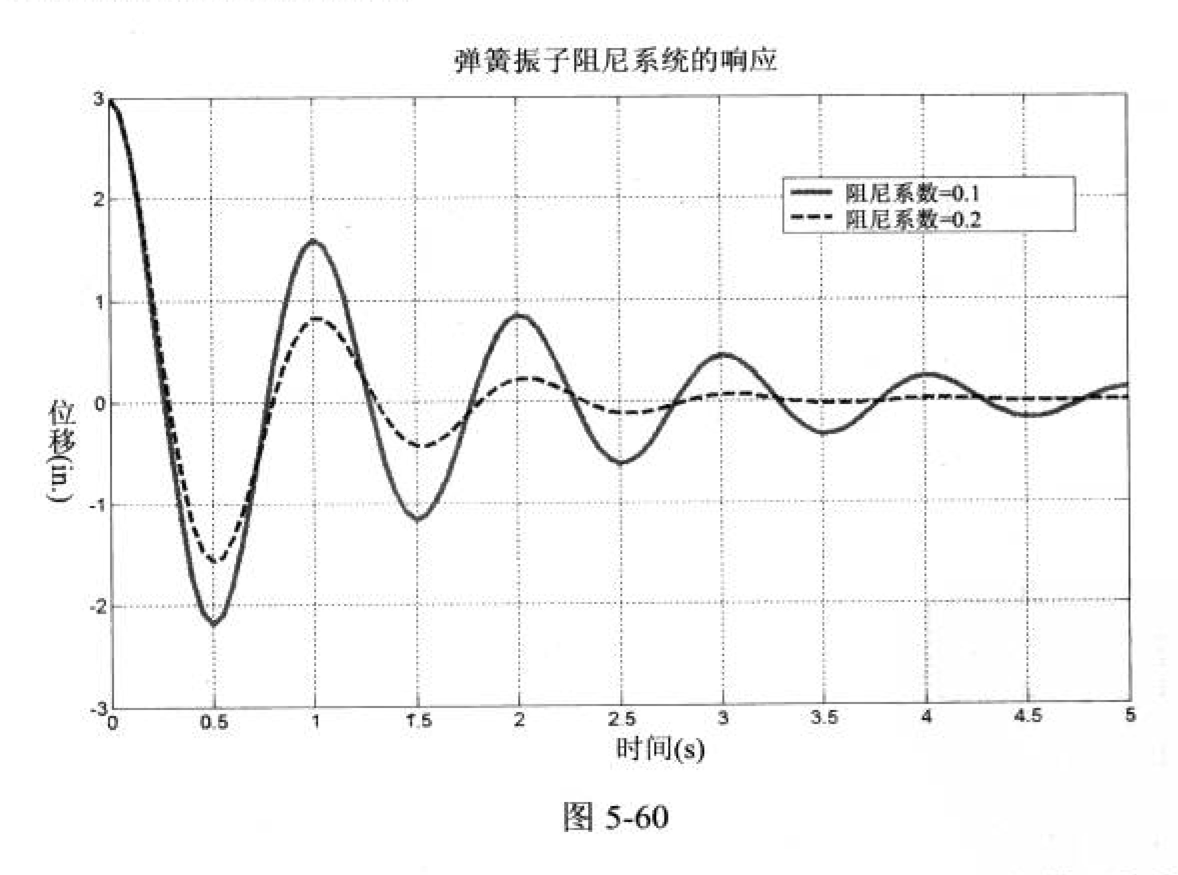


图 5-60

MATLAB 另一个有用的绘图命令是 fplot。要用 fplot 绘图，必须把需要绘制的函数定义为 m-file。fplot 命令的一般格式是：

$$\texttt{fplot('functionname',[}x_{lower}\ x_{upper}\texttt{])}$$

这里的 x_{lowewr} 和 x_{upper} 是函数参数的上下限值，即它们定义了自变量的取值范围。除了用户定义函数，fplot 命令也可以用来绘制 MATLAB 内置的函数。例如，我们绘制 sin 函数在 0~2π 之间的曲线，要输入以下命令：

```
>> fplot('sin',[0,2*pi])
```

要绘制弹簧振子阻尼系统的位移曲线，必须先建立一个函数文件。

在编辑里建立建立以下文件，为了节省时间，读者可以从 SMD 文件复制部分代码：

```
1 function y = displace(t)
2 y0 = 3.0;
3 dc = 0.10;
4 fr = 2*pi;
5 fd = fr*sqrt(1-dc^2);
6 c = cos(fd*t);
7 s = sin(fd*t);
8 e = exp(-dc*fr*t);
9 y = (y0*c + y0*dc*fr/fd*s)*e;
```

把这个文件保存为 displace。求这个函数在 2s 时的值，验证它是否正确。

```
>> displace(2)
ans =
     0.8467
```

输入下面的命令，绘制这个位移函数在 0~5s 的曲线：

```
>> fplot('displace',[0 5])
```

最后得到的曲线如图 5-61 所示。

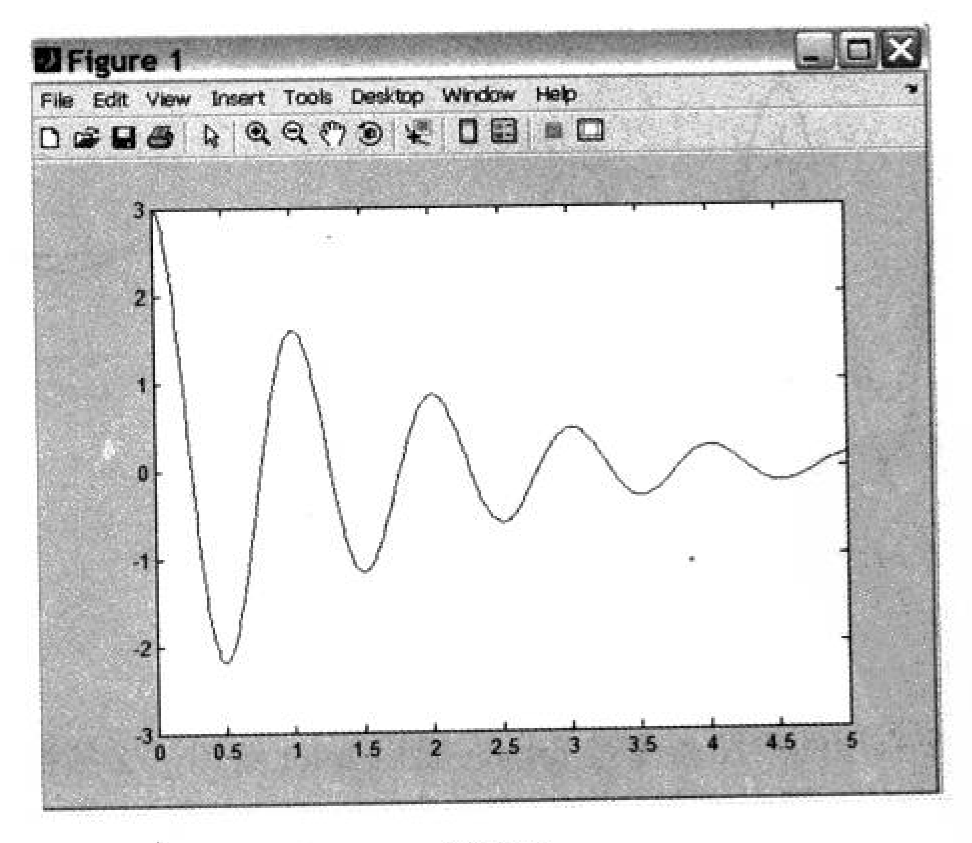


图 5-61

对于给定的初始位移、阻尼系数和固有频率值，从这个函数可以得到满意的位移随时间变化的曲线。但是，如果我们想比较系统在不同的阻尼系统时的位移曲线，则不得不建立另一个以阻尼系数为输入参数的函数。更好的办法是建立多个参数的函数。这个函数有 4 个输入参数：初始位移、阻尼系数、固有频率和时间，输出是系统的位移。

修改这个函数程序的第一行，把它改成如下的内容，并把它保存为 displace2：

```
1 function y = displace2(y0,dc,fr,t)
```

为了测试这个函数，我们把它的 4 个输入参数设置为 y0=3，dc=0.1，*fr*=2π 和 *t*=2：

```
>> displace2(3,.1,2*pi,2)
ans =
     0.8467
```

为了用 fplot 命令绘制这个多输入参数的函数，除了自变量 *t* 设置取值范围外，我们必须给其他 3 个参数设定确定的值。

在命令行输入以下命令：

```
>> fplot('displace2(3,.1,2*pi,t) ',[0 5])
>> hold on
>> fplot('displace2(3,.2,2*pi,t) ',[0 5])
```

注意在这个函数的参数列表里，插入一个变量名(t)作为函数的自变量，并在其后用方括号表示自变量的取值范围。与前面一样，为了便于比较阻尼系数不同对振动的影响，我们在同一个图里绘制两条曲线。最后得到的曲线如图 5-62 所示。现在就可以用交互的编辑工具修改这个图形了。

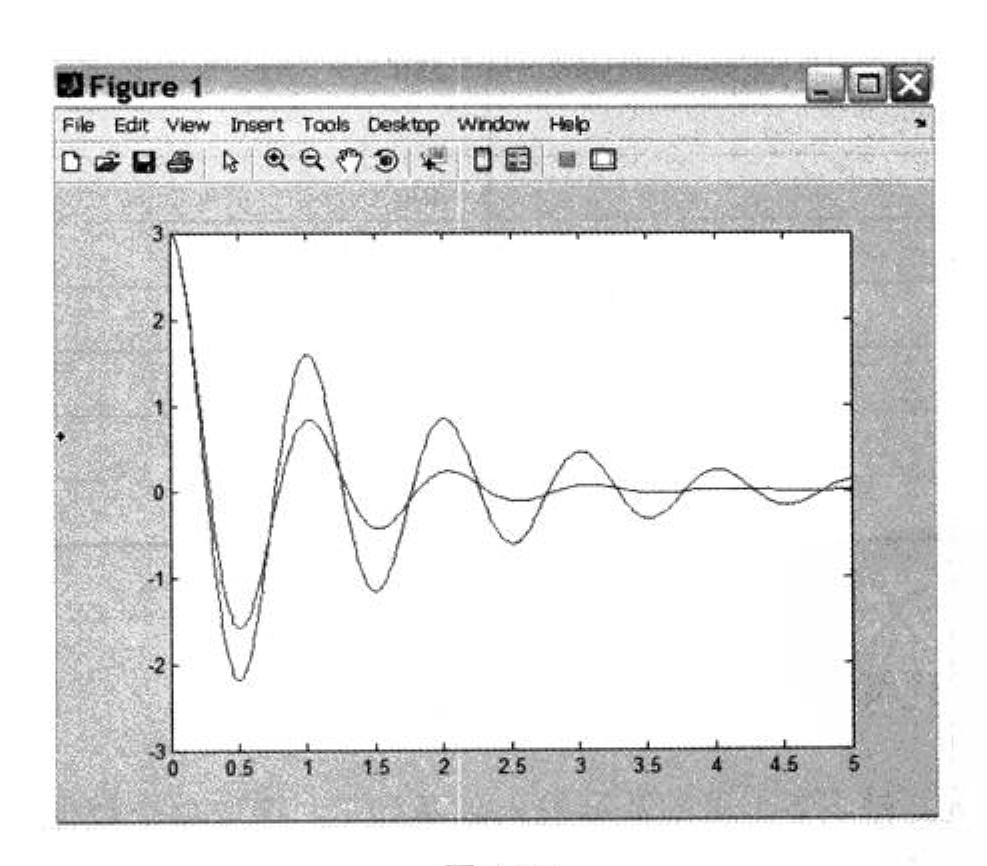


图 5-62

5.2.3 在 Excel 里绘制数据和拟合曲线

现在我们讨论一个垂直弹簧的受力测试。这个弹簧系统如图 5-63 所示。弹簧夹在两个平板之间，外力逐渐地作用于上平面上，引起弹簧压缩。当外力达到 25N 时，把此时的上平板位置设置为 0。设当上面的平板移动距离 *d*(单元毫米)时，作用于上平板上的力为 *F*(单位为 N)，记录 *d* 和 *F* 的值，当上平板移动 10mm 时，测试结束。表 5-1 就是这个试验的记录数据。如果我们绘制出作用力与位移的曲线，就可以求得它的斜率，这个斜率就是弹性常量。

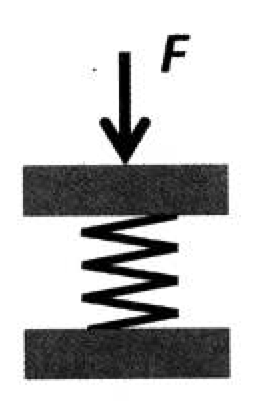


图 5-63

表 5-1 弹簧测试数据

d(mm)	*F*(N)
1	102
2	156
3	230
4	317
5	400
6	452
7	530
8	611
9	670
10	739

例 5.5

绘制表 5-1 里的弹簧测试数据，并求出它的弹性常量。

解：

新建一个工作表，把表 5-1 里的数据输入到两列里。选取整个数据域，从 Ribbon 中选择【插入】|【散点图】)|【仅带数据标志的散点图】命令，如图 5-64 所示。把这个图表移动到工作簿的另外一个工作表里，并根据读者的要求设置它的标题、坐标标签和图形格式。

最后得到的结果如图 5-65 所示。



图 5-64

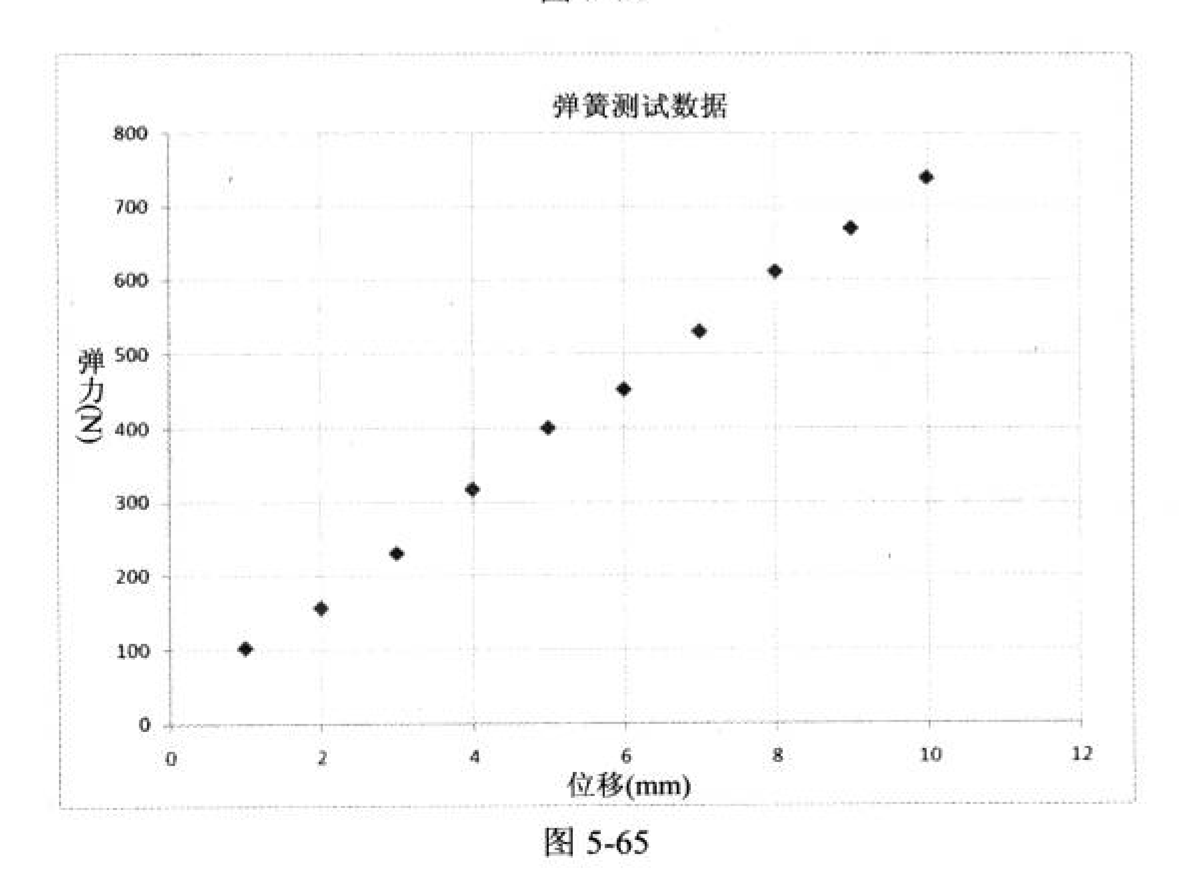


图 5-65

读者可以手工画一条直线，连接第一个点和最后一个点，但是我们希望找出与这些数据拟合得最好的直线。因此，我们准备采用最小二乘法拟合。所谓最小二乘法拟合，是指各数据点的 y 值与由直线方程计算得到的 y 值差的平方和最小。由于两者之差再经过平方，因此它们差值正负不会相互抵消。利用微积分公式，可以求得直线的斜率和 y 轴的截距。Excel 和 MATLAB 都有相应的命令计算最小二乘拟合值。

右击其中一个数据点，在弹出的快捷菜单中选择【添加趋势线】命令，如图 5-66 所示。在弹出的【设置趋势线格式】对话框中的【趋势线选项】选项卡中，将拟合类型选择为【线性】，并选择【显示公式】和【显示 R 平方值】两个选项，如图 5-67 所示。单击【关闭】按钮。

趋势线的类型不止一种，后面的教程还将使用指数类型，趋势预测选项根据第一个点和最后一个点外插得到，但是读者要谨慎使用这个选项。当我们在测量数据之外进行外插，就无意中做了这样的假定：影响测量范围内数值的因素也影响测量范围之外的数据。此外

还有一个选项，把 y 轴的截距设置为某个值(例如，当我们希望这个直线经过原点)。

选取图上的方程，修改它的字体大小。如有必要，右击方程的文本框，在弹出的快捷菜单中选择【设置趋势线标签格式】命令，可改变方程文本框的填充颜色和边框颜色。

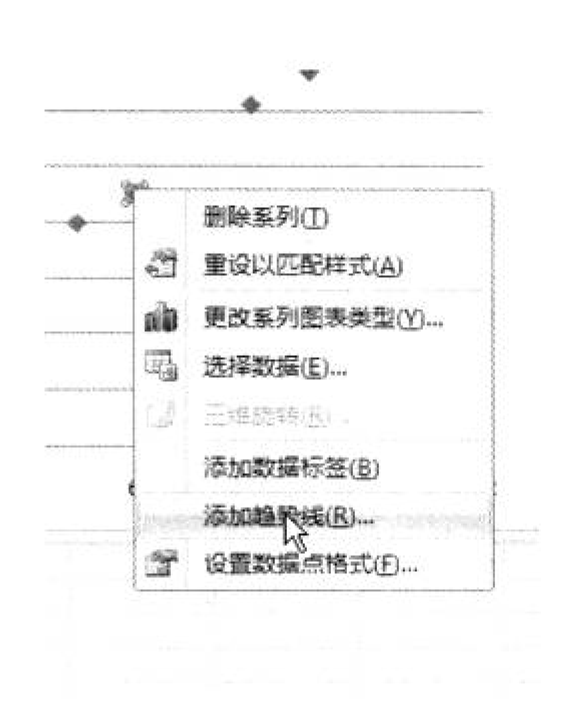


图 5-66

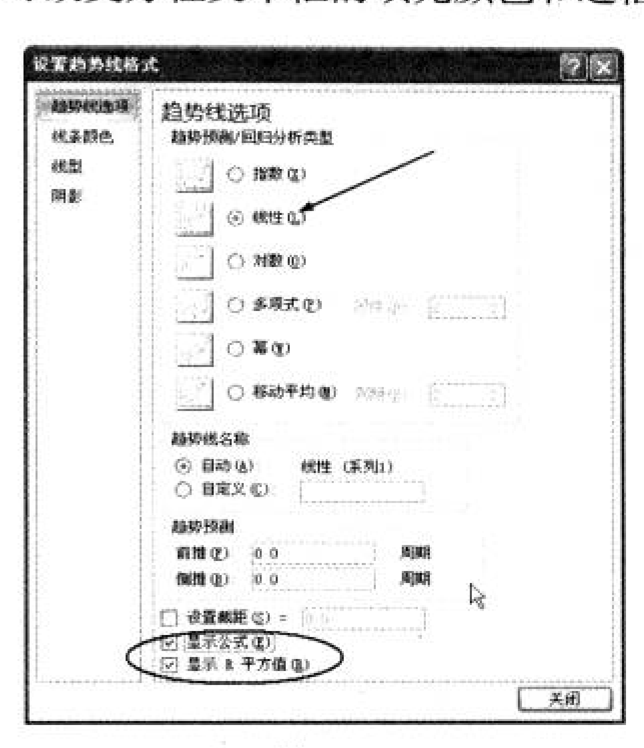


图 5-67

得到的拟合曲线如图 5-68 所示。R^2(称为相关系数)值用来衡量曲线的拟合程度。这个值越接近 1，直线与数据拟合程度就越高。在这个例子里，根据理论可知，它们之间存在线性关系，因此，这个值非常接近 1。在其他情形下，由于有多个自变量影响结果，因此相应的相关系数就可能比较小。在这些情况下，相关系数常用来决定哪个自变量对结果影响最大。

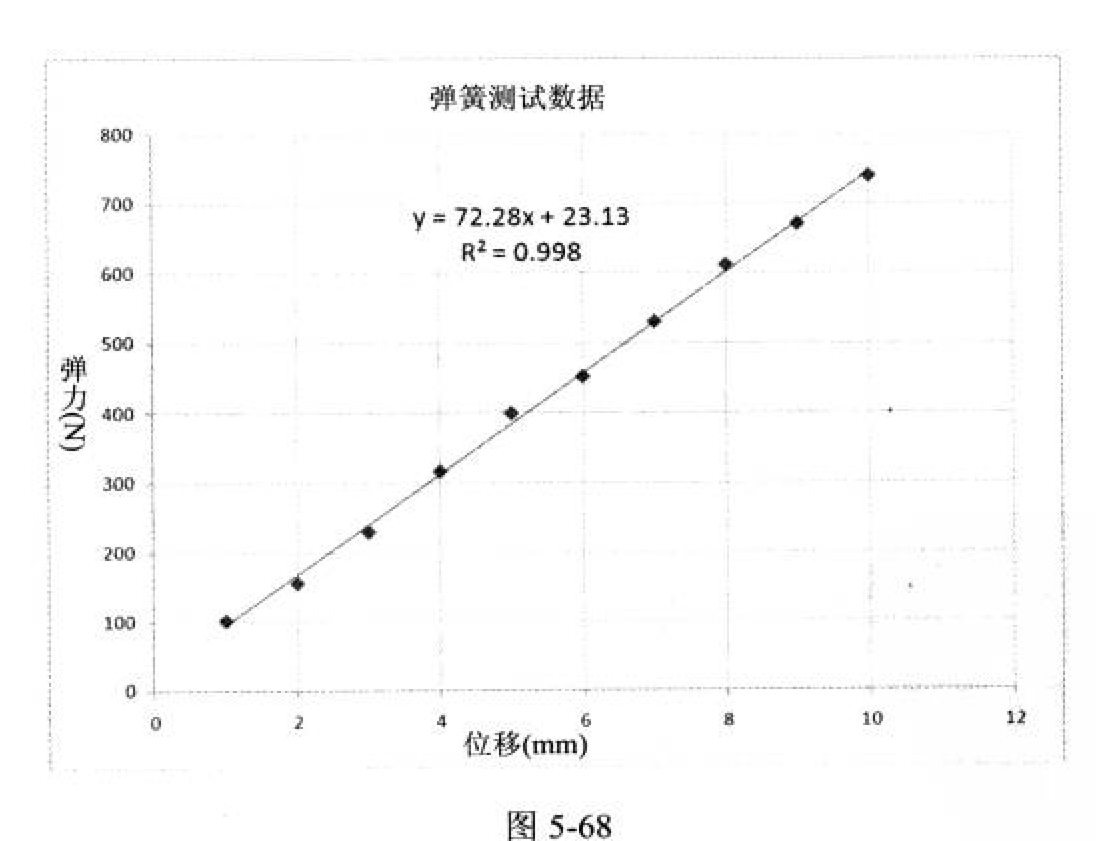


图 5-68

从图 5-68 可知，弹簧的弹性常量即方程的斜率为 72.3N/m。

例 5.6

把一杯水放在微波炉里加热，当水快沸腾时，停止加热，并从微波炉里拿出这杯水。在这杯热水里插入温度计，当水的温度冷却到 200℉时，启动计时表。每隔几分钟，记录水温，时间-温度数据如表 5-2 所示。假如室内温度为 77℉。用测试数据，绘制温度数据点，根据这些数据点拟合一条曲线。

表 5-2　冷却实验数据

时间(min)	温度(°F)
0	200
1	192
2	185
3	179
4	174
5	169
10	151
15	138
20	126
25	117
30	106
60	84

解：

在 Excel 的工作表里输入表 5-2 里的数据。建立散点图，并把它设置为【有数据标志，无线条】，如图 5-69 所示。

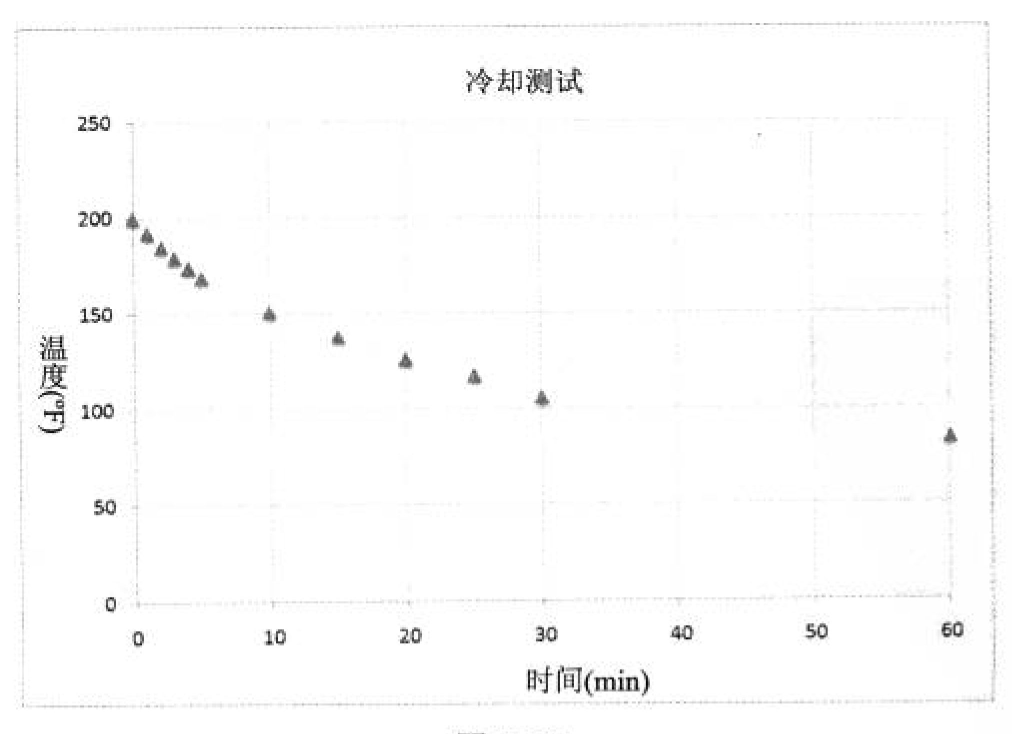


图 5-69

读者可能已注意到，数据点并不是均匀分布的。许多学生常犯的错误就是选择直线而非散点作为图形类型。选择直线图会把数据点沿 x 轴等间隔分布，而不管数据值如何。如果 x 数据属于分类数据，则使用直线图是比较合适的。反之，如果 x 数据是数值型，则不要使用直线图。

从图上很容易看出，这些数据点并没有沿着一条直线分布。如果读者回到图 5-67，应该知道，Excel 允许我们使用其他几类曲线拟合数据。其中两类趋势图，如果我们把它们应用到这里的数据，就会出错。对数曲线，包含一项 $\ln(x)$，当 x=0 时，这一项值没有定义。幂运算型曲线可以取如下的形式：

$$y = Ax^b$$

我们可以看出，这条曲线肯定会经过原点，因此，指数曲线也不适合于这里的数据。"移动平均"曲线从技术上根本不是曲线，而是数据光滑选项。这样，只剩下两种可能：多项式趋势线和指数型趋势线。

对于多项式曲线，我们必须确定多项式的次数。图 5-70 的趋势曲线是由二次多项式生成的，因此该曲线具有以下形式：

$$y = Ax^2 + Bx + C$$

多项式的次数越高，拟合曲线越好。图 5-71 是 4 次多项式拟合曲线：

$$y = Ax^4 + Bx^3 + Cx^2 + Dx + E$$

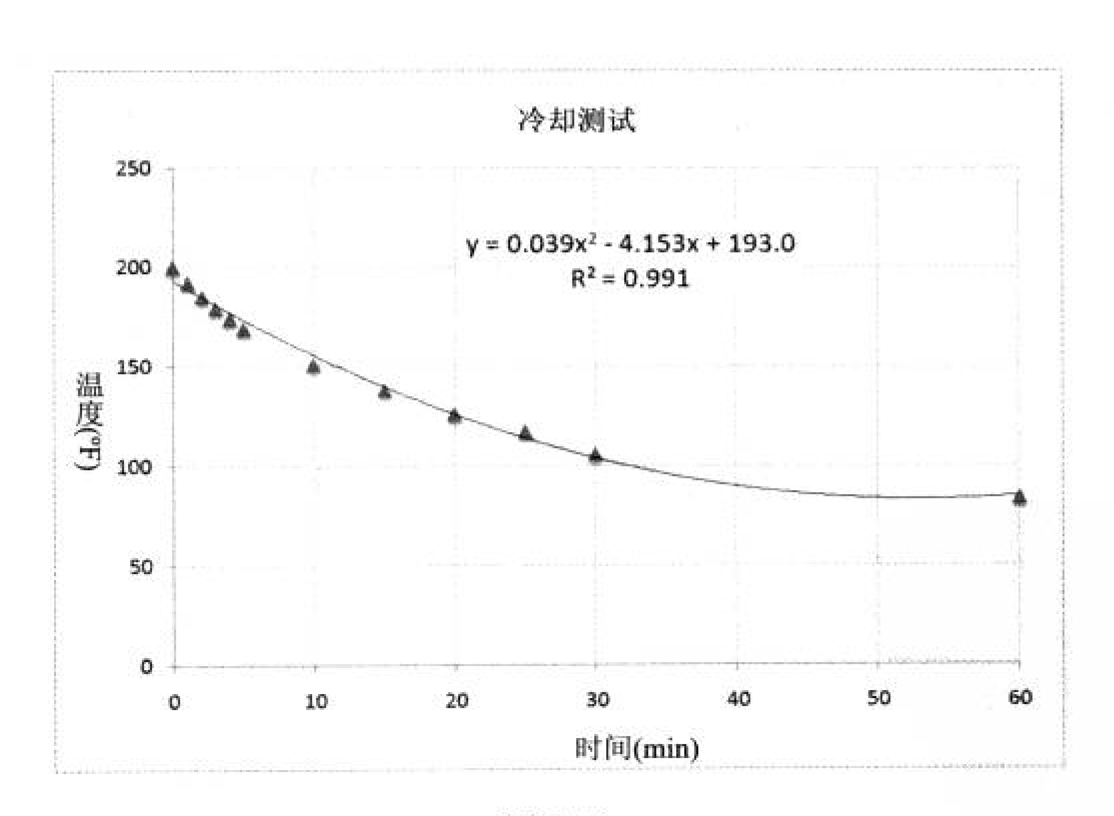


图 5-70

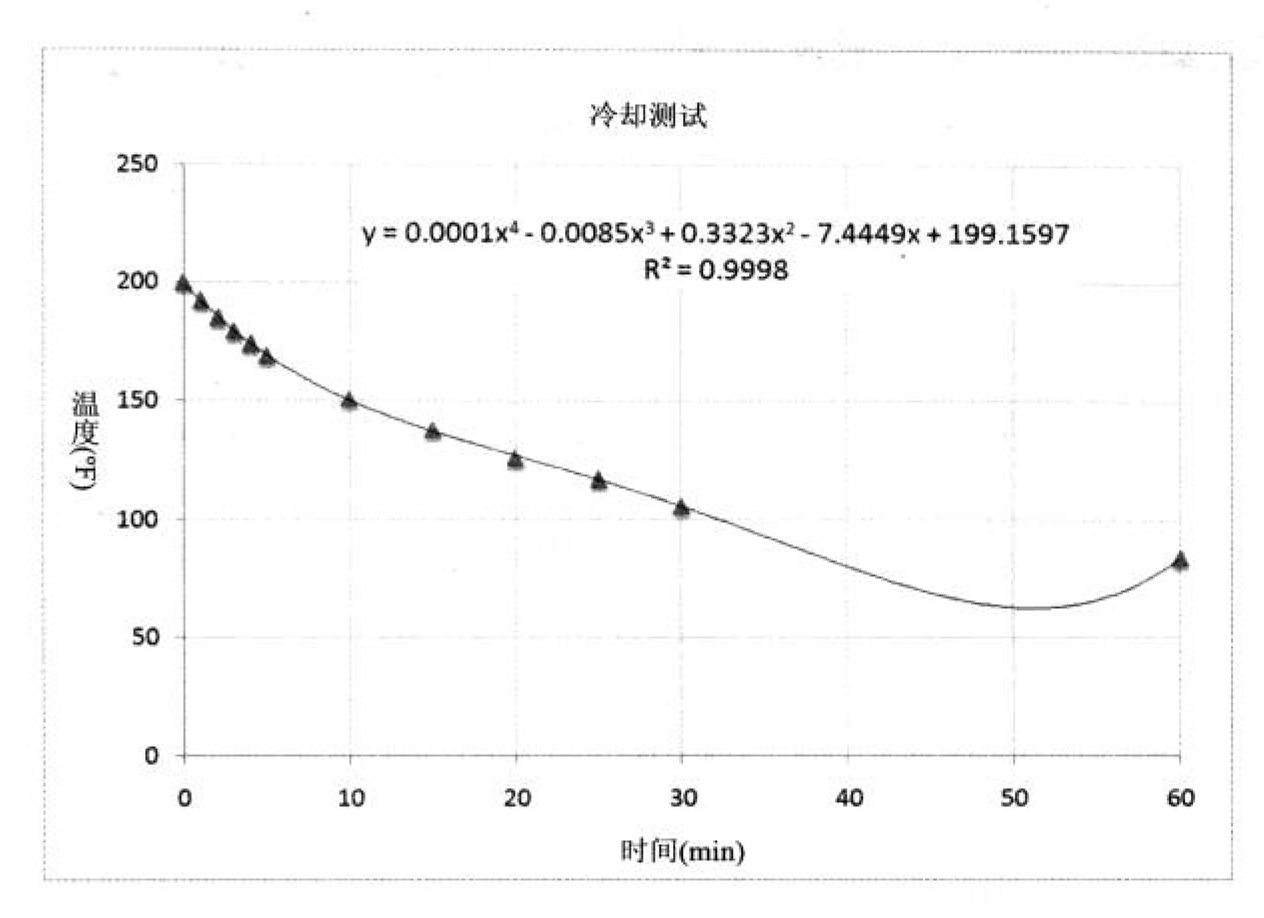


图 5-71

这两个拟合曲线的 R^2 都很接近 1，因此，它们都很好地拟合了这些数据。用更高次多项式可以得到更好的拟合效果。但是这些多项趋势曲线都存在一个问题，特别是 4 次多项拟合曲线更加明显。趋势曲线的预测结果表明，在一定时间之后，温度随着时间会升高。从这个事实出发，我们得到以下一个非常重要的规则，读者在绘制拟合曲线时必须记住：不管曲线与数据的拟合效果如何，曲线必须与数据的实际性质一致！

显然可见，这个模型——水温冷却后再升高——是没有实际价值的。

现在只剩下指数趋势线。使用指数曲线是有理论根据的，因为，指数模型经常用来模拟生长、衰减等问题，在这些问题里，生长率是递增或递减的。其中几个熟悉的例子是，放射性物质的衰减，或银行存款的复利计算。在本章的第一个教程里，弹簧振子阻尼系统的位移就是按指数衰减的(在振动过程中)。现在我们就把指数趋势线应用于这一组数据上。

右击某个数据点，在弹出的快捷菜单中选择【添加趋势线】命令，在弹出的【设置趋势线格式】对话框中，选择【趋势线选项】选项卡中的【指数】类型，选择【显示公式】和【显示 R 平方值】两个选项，如图 5-72 所示。单击【确定】按钮。

右击趋势图上的标签框(公式与 R^2 值的文本框)，改变字体大小、填充颜色和边框颜色。另外把数值格式从【常规】改为【数值】，并且将小数位数设置为 5 位，如图 5-73 所示。

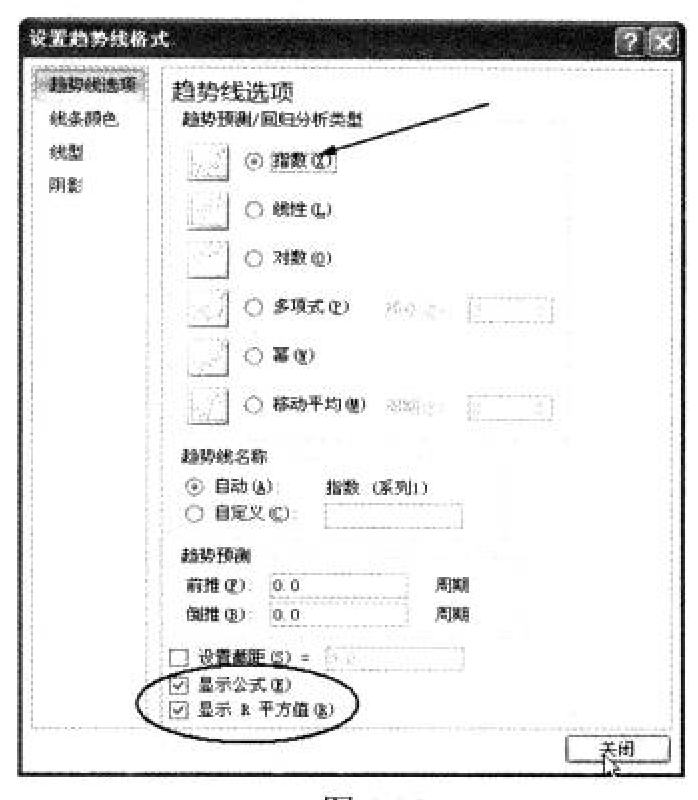


图 5-72

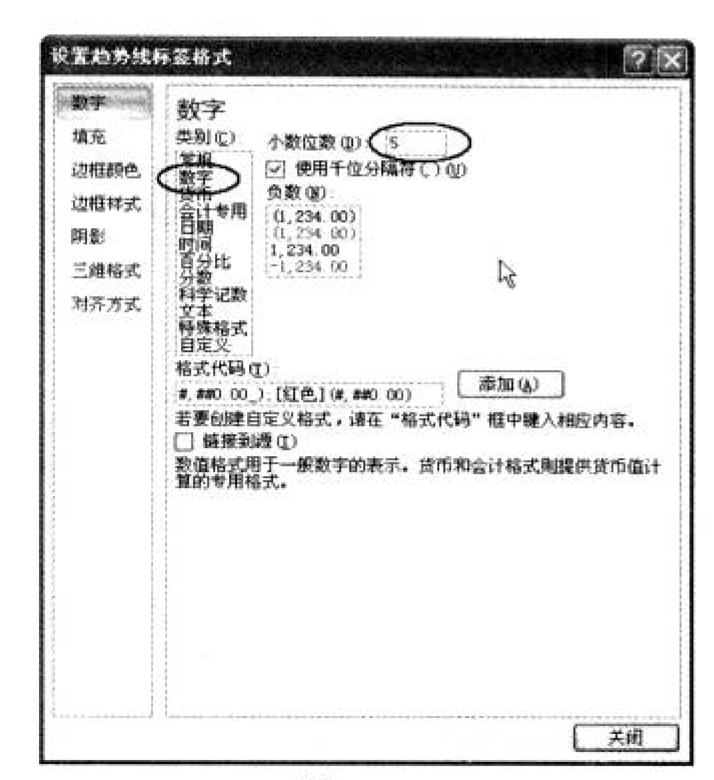


图 5-73

从表面上看，没有必要使用 5 位小数，但是指数是一个非常小的数。在曲线的方程上只用两位或三位小数可能会影响它的精度。

最后得到的趋势曲线如图 5-74 所示。正如前面曾提到的，指数曲线总的来说是比较合适的，但是它与数据吻合得不是很好。同样要考虑到实际问题，当水冷却时，温度最终降到室温左右(在本例中是 77℉)，但是根据我们的指数方程预测结果，当时间 t 变得很大时，温度接近于 0ºF。换言之，即使室温是 77℉，水也会结冰。而正确的模型能够预测到水温与室温之差趋近于 0ºF。

在这个工作表中添加一列，输入一个公式，它把水的温度减去 77，图 5-75 显示了两者温度之差。右击图表区域，在弹出的快捷菜单中选择【选择数据】命令，如图 5-76 所示。

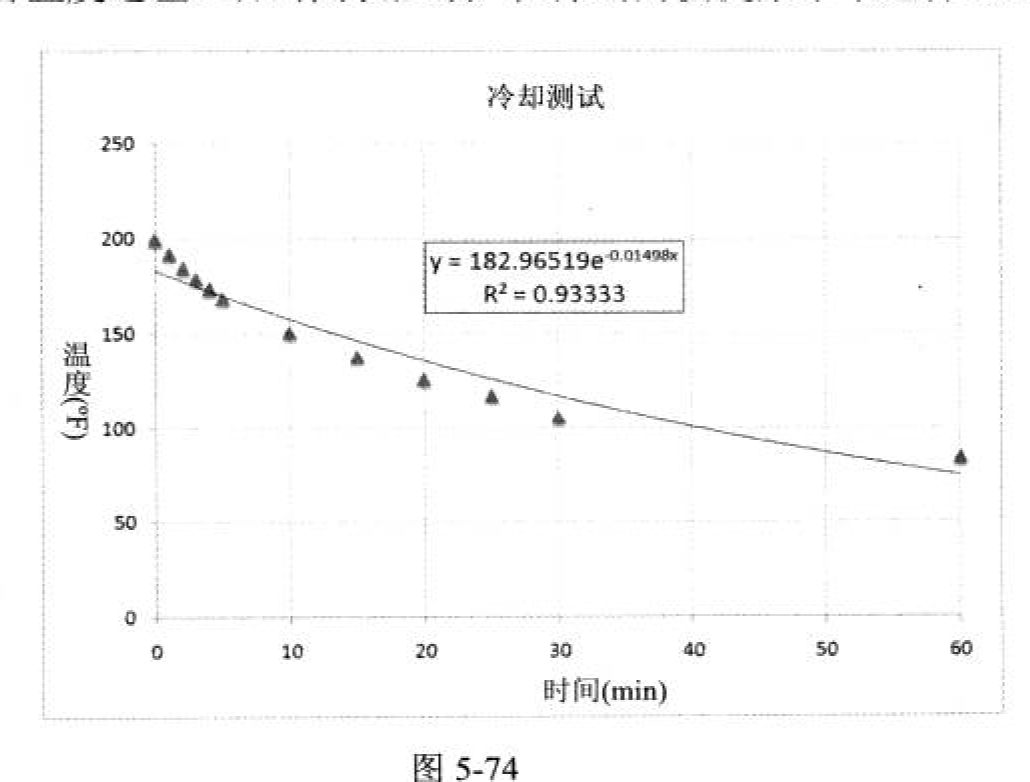


图 5-74

	A	B	C
1	Time, minutes	Temperature, deg F	Delta T, deg F
2	0	200	123
3	1	192	115
4	2	185	108
5	3	179	102
6	4	174	97
7	5	169	92
8	10	151	74
9	15	138	61
10	20	126	49
11	25	117	40
12	30	106	29
13	60	84	7

图 5-75

在弹出的【选择数据源】对话框中，选择【系列 1】，然后单击【编辑】按钮，如图

5-77 所示。在弹出的【编辑数据系列】对话框中，单击【Y 轴系列值】文本框右边的图标，如图 5-78 所示，选择包含温度差的单元区域，如图 5-79 所示。

把 y 轴的标签改为“Temperature Difference,degrees F”。

最后得到的结果如图 5-80 所示。注意，趋势线与数据吻合得非常好，而且当时间增大时，温度差也趋近于 0，这也是符合情理的。

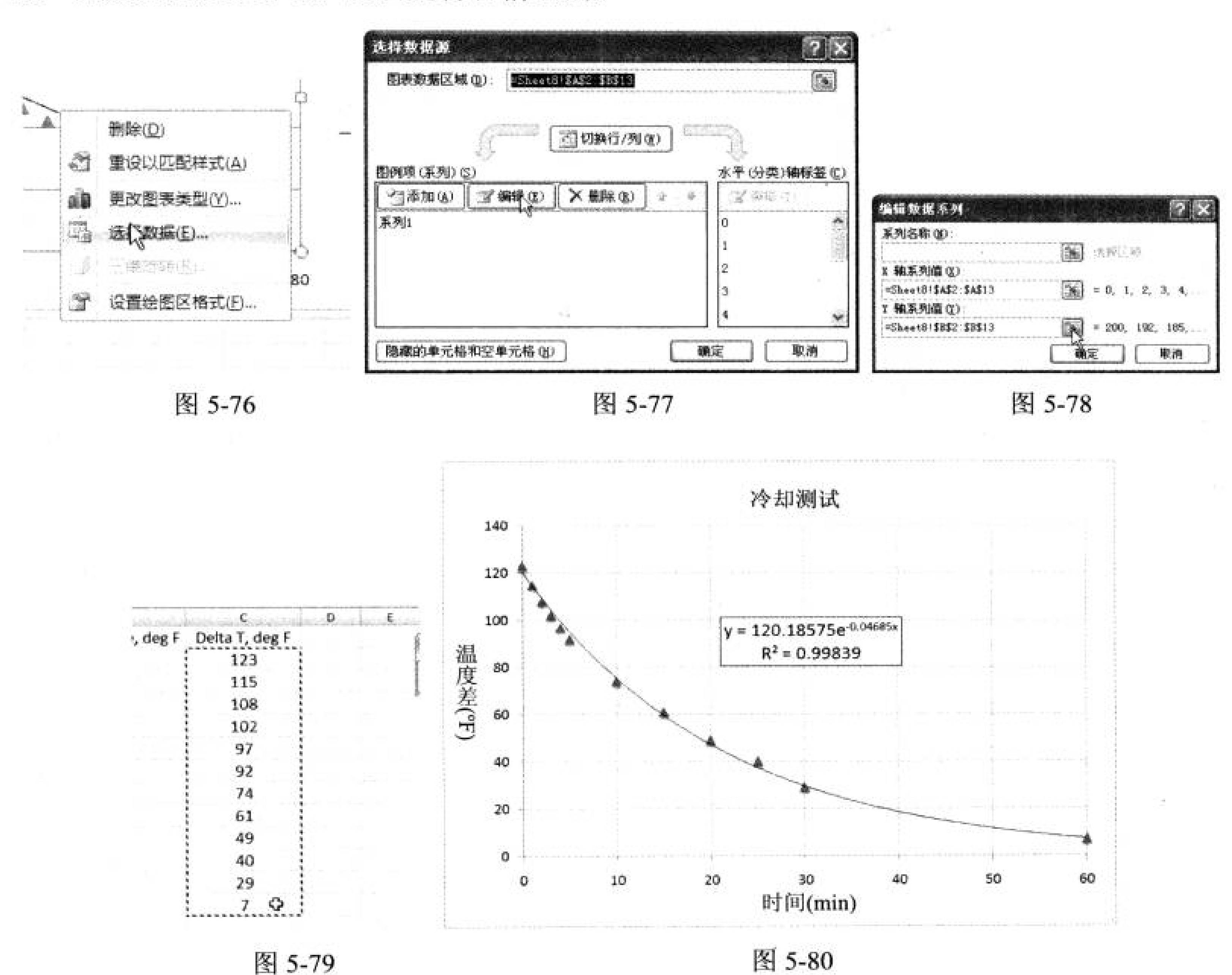


图 5-76　　图 5-77　　图 5-78

图 5-79　　图 5-80

5.2.4　在 MATLAB 里绘制数据和拟合曲线

与 Excel 一样，MATLAB 也有许多曲线拟合选项。下面这个例子说明 MATLAB 绘制拟合曲线的步骤。在这个例子里，我们先绘制前面 Excel 例子里的弹簧测试数据，并根据这些数据，得到一条拟合曲线。

例 5.7

绘制表 5-1 中的弹簧测试数据，并求出它的弹性常量。

解：

在 MATLAB 里，把数据保存到两个数组里。由于位移数据是等间隔的，因此我们可以用冒号运算符生成这个数组。下面绘制这个数组。

```
>> d = 1:10;
>> F = [102 156 230 317 400 452 530 611 670 739];
>> plot(d,F)
```

得到如图 5-81 所示的曲线。

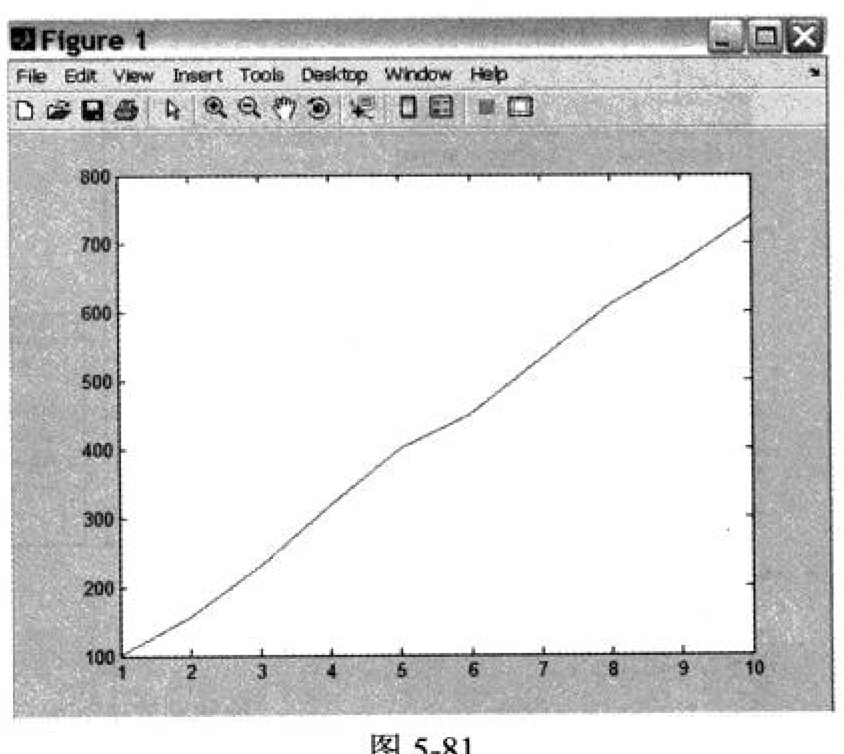


图 5-81

为了编辑图形，单击工具栏上的箭头图标。右击曲线，选择 Marker(数据标志)|Square(正方形)命令，如图 5-82 所示。再次右击，选择 Line Style(线型) | none(元)命令。

从主菜单选择 Tools(工具) | Basic Fitting(基本拟合)命令，如图 5-83 所示。在弹出的 Basic Fitting 对话框中选择 Linear(线性)拟合类型。然后选择 Show equations(显示方程)选项，并把 Significant digits(有效数字位数)设置为 4，如图 5-84 所示。单击 Close(关闭)按钮。

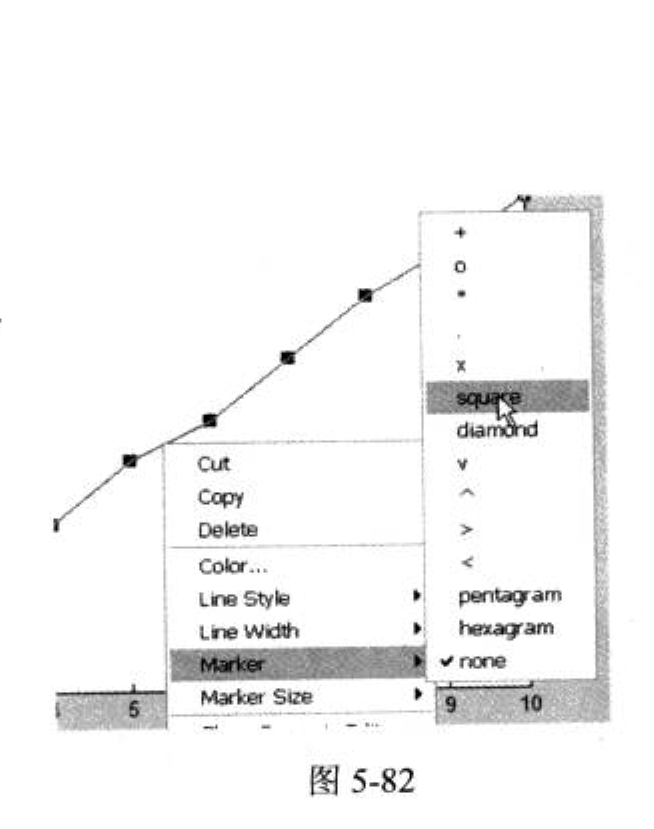


图 5-82

图 5-83

需要说明的是，如果读者希望不需要在图形上显示斜率和截距就可以看到它们，则单击 Basic Fitting(基本拟合)对话框右下角的箭头，就会看到这些参数的值。

修改这个图形，添加网格线、轴标签和图标题。

最后得到的图形如图 5-85 所示。从直线的方程我们可以看出，该弹簧的弹性系数为 72.3N/mm。

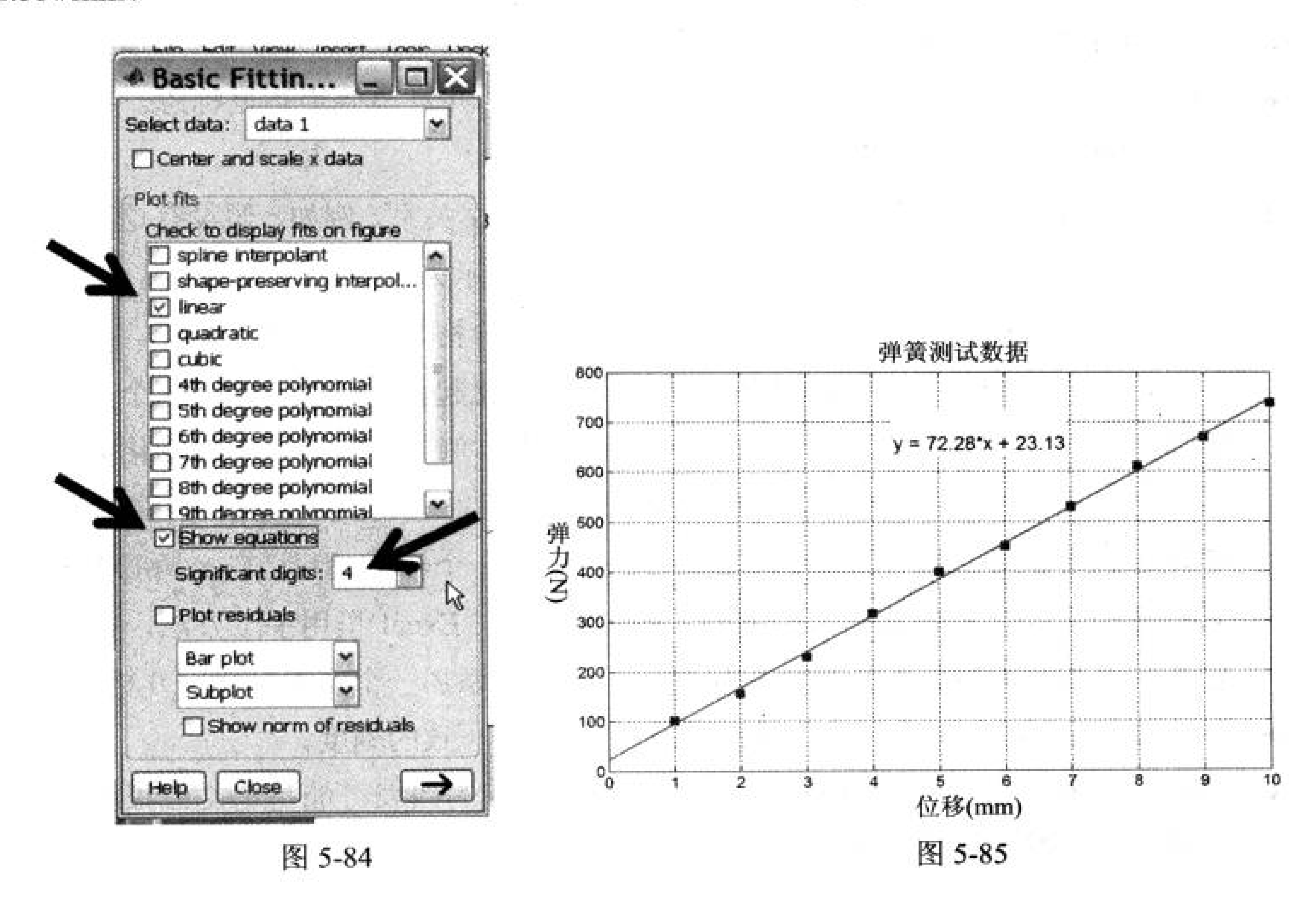


图 5-84　　图 5-85

5.3　图表制作指南

当我们需要建立一个用在论文、报告、演示文稿或作业中的图表时，为了使生成的图形容易理解和阅读，必须遵循以下几个规则：

- 给图表制定一个描述性的标题。标题在图形的顶部或底部。如果文章使用图号，标题必须在图号之后。
- 给两个坐标设置标签，并说明单位。
- 选择比较大的字体，以方便阅读，特别当需要缩小图形时，更是如此。
- 网格线很有用，但是网格线的颜色必须比曲线本身的颜色浅一些，而且线型较细。
- 当一个图表显示多个数据曲线时，必须使用图例或者在曲线附近加上文字标签。颜色是区别不同曲线的最好办法。但是如果图表需要打印或转换为黑白图像，则

白色图例没有多大作用。如果图表有两三条曲线，可以考虑在曲线旁边使用文字标签，因为在这种情况下图例是不好理解的。

- 当需要显示实际数据点时，必须用数据标志显示这个数据点。这些数据点经常用直线段连接起来，除非需要显示一条理论曲线或趋势线。一个例外情形是，当数据点非常多，以至于数据标志会造成混乱的感觉。在这种情况下，只需要显示曲线。曲线是由两个数据点间的线段拼接而成的。
- 当需要绘制函数的曲线时，最好使用光滑曲线，并且不要用数据标志。在这种情况下数据点本身就没有特别的意义。
- 在决定 y 轴的刻度单位时，需要考虑图表的上下文关系。例如，假设有这样一种情形：某个公司把一周之内股票每天的收盘价格显示为图 5-86 和图 5-87。在图 5-86 中，由于 y 轴坐标从 0 开始，因此，本周的股票价格显得相对比较平稳。在图 5-87 中，y 轴使用自动刻度，使得每天的股票价格变化起伏比较大，到底哪个正确呢？就本例而言，图 5-87 是典型的股票价格图。在本例中，股票价格的变化需要予以重视，而图 5-86 没有很好反映股票价格的易变性。有时我们使用"截断刻度"说明坐标轴的刻度可以伸展到 0 刻度。图 5-88 就是一个例子。但是，Excel 和 MATLAB 都没有内置的功能建立这一类图表。图 5-88 是在 Excel 里用手工办法，在 y 轴上添加一个白色方框，在方框里添加两条短线，再添加一个文本框表示 0 标签。
- 当需要显示两个图表用于比较时，它们必须使用一致的刻度。
- 最后，虽然在工程论文或报告中，一个设计良好的图表可能是论文或报告的一个重要的、有时是必不可少的内容，但是必须在正文中引用这个图表。

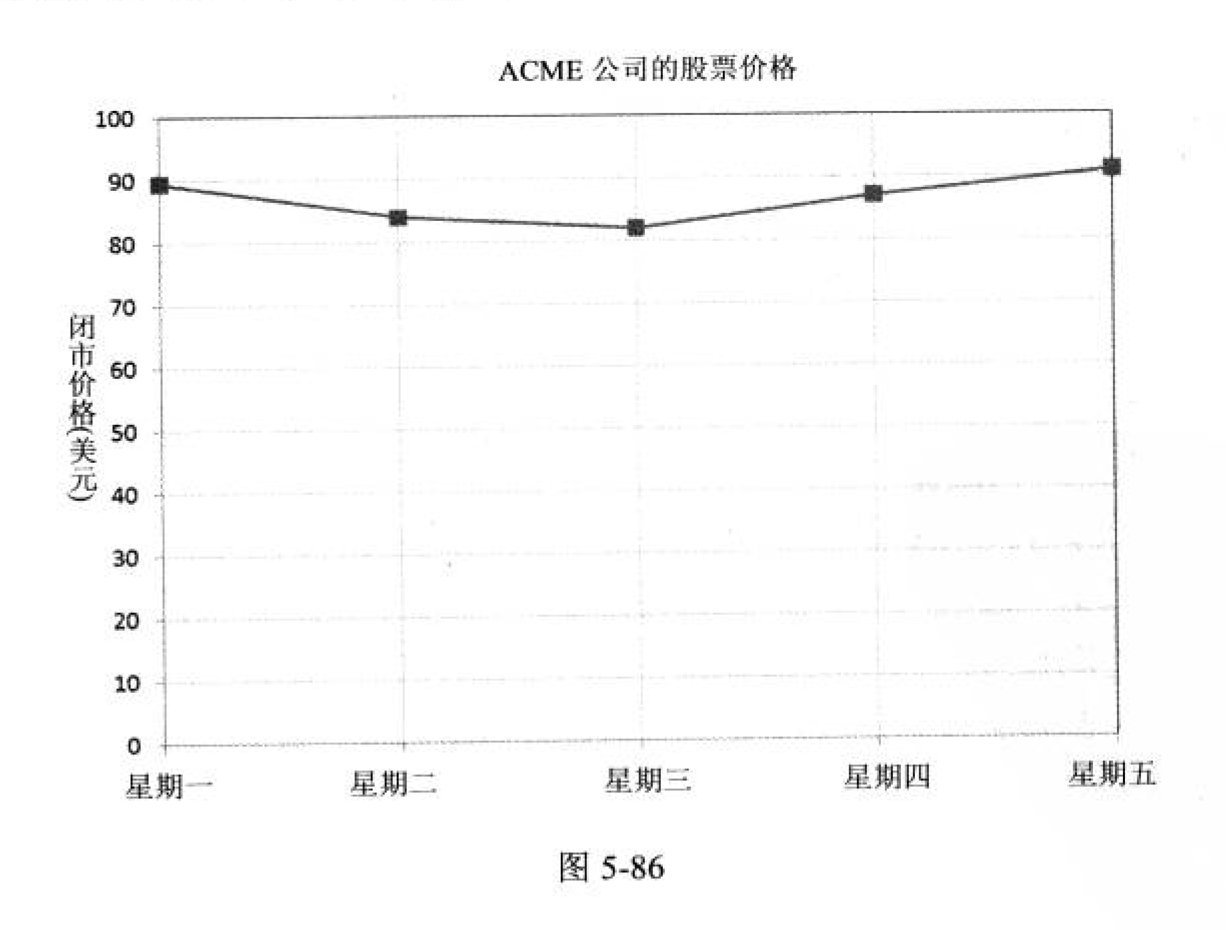


图 5-86

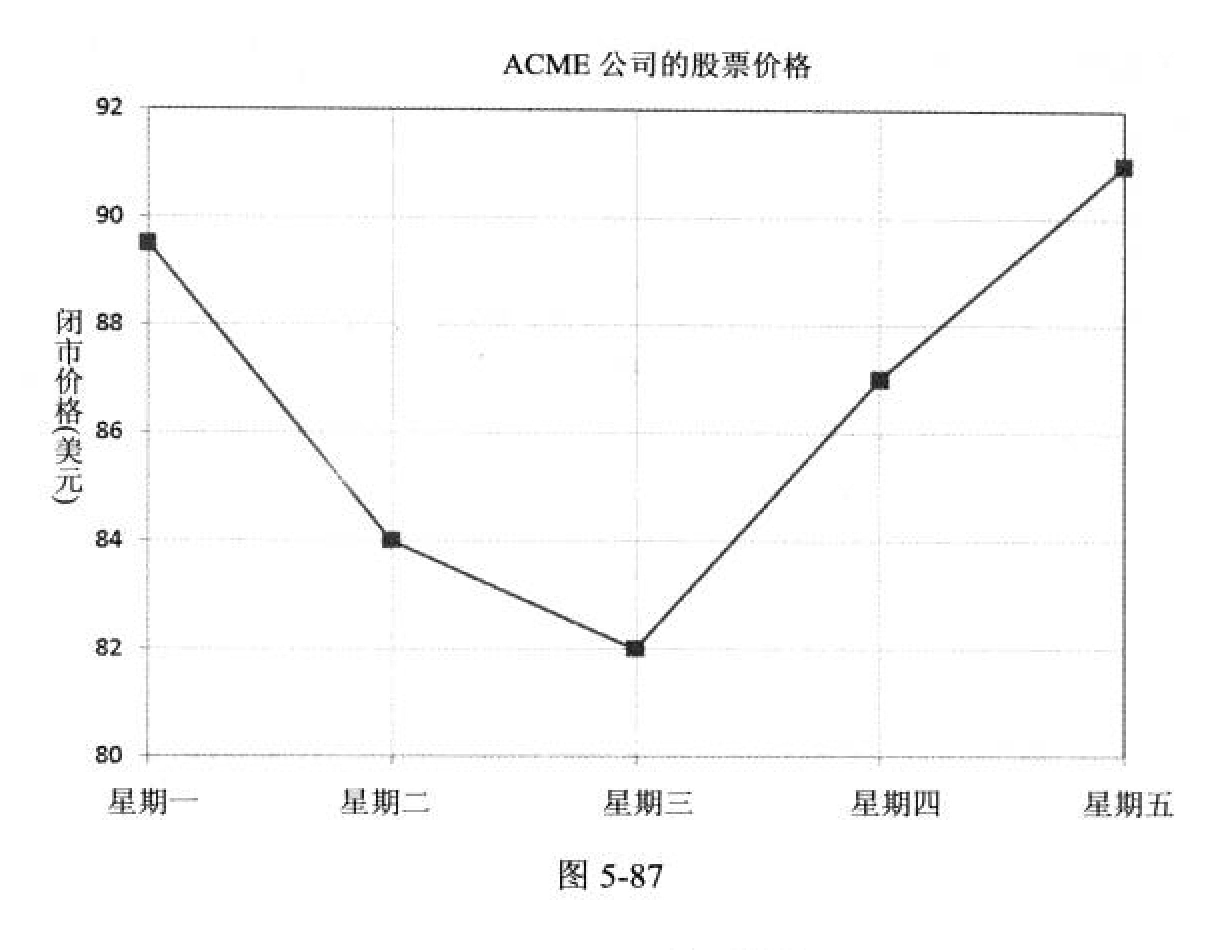


图 5-87

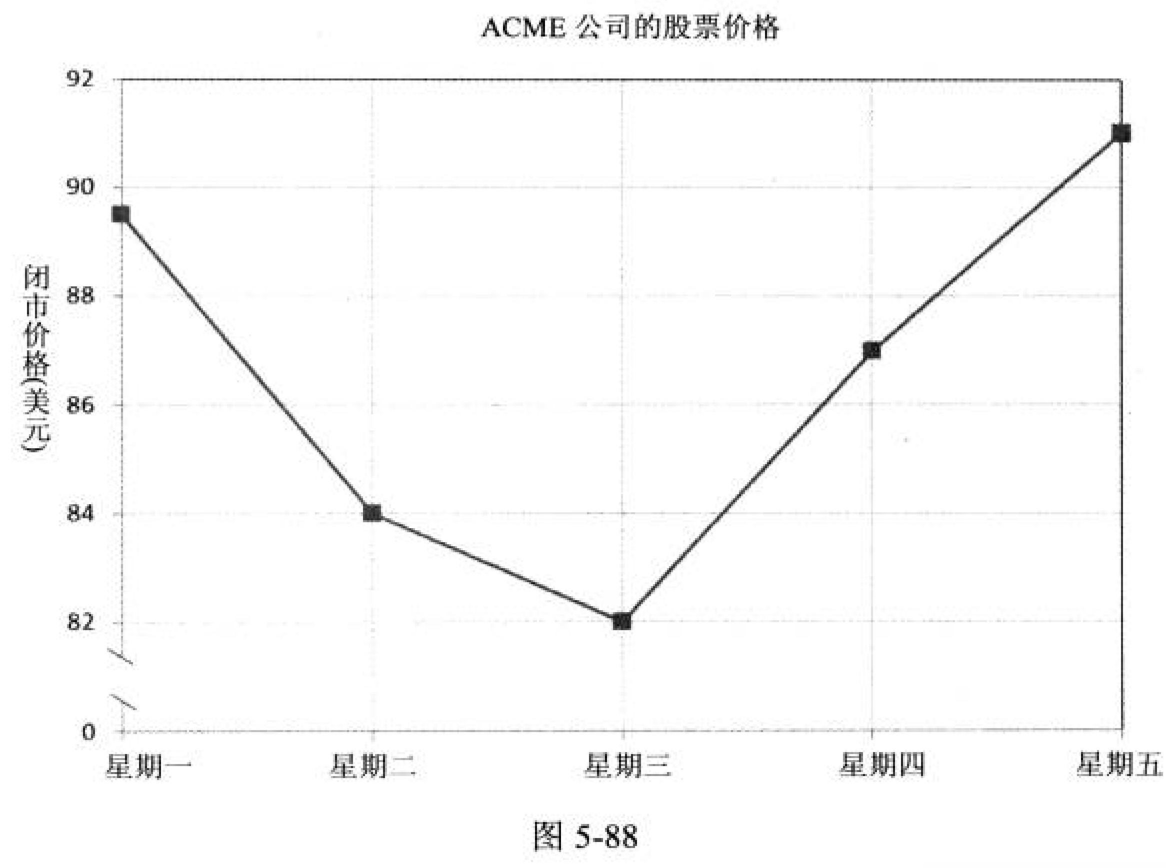


图 5-88

5.4　教程：用 Excel 建立其他类型的图表

前面曾提到，散点图(XY)是当前工程中使用最广泛的一类图表，然而，我们也经常用到其他类型的图表。这一节我们将介绍如何在 Excel 里建立饼图、柱形图和 Pareto 图。在 MATLAB 里也可以建立这些类型的图表。然而，通常 Excel 是工程人员的首选工具，因为

在 Excel 里输入数据比较容易，修改图表比较灵活。

在本节的几个例子里，我们将分析某个产品(用产品 ABC 表示)的制造成本。表 5-3 列出了该产品在前一年和当年的制造成本。

表 5-3 产品 ABC 的制造成本

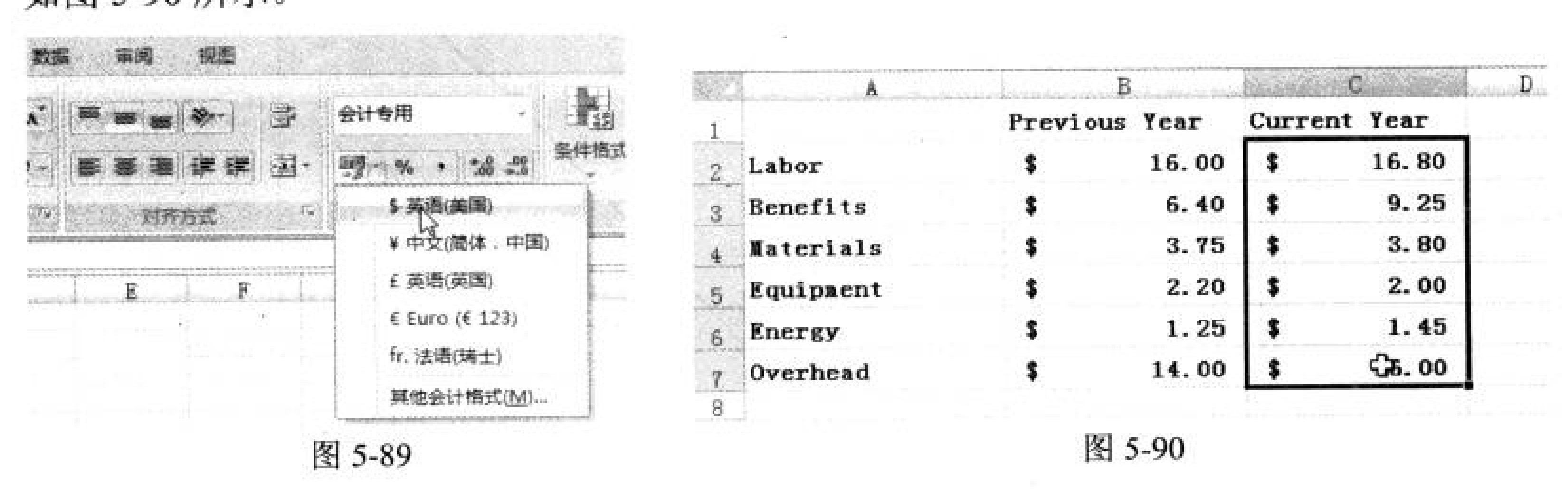
	前一年	当年
直接劳动力成本	¥16.00	¥16.80
利润	¥6.40	¥9.25
材料	¥3.75	¥3.80
设备	¥2.20	¥2.20
能量	¥1.25	¥1.45
管理费用	¥14.00	¥16.00

例 5.8

根据表 5-3 的当年制造成本绘制一个饼图。

解：

建新一个工作表，按表 5-3 输入数据。把包含美元值的单元设置为会计专用格式。方法是，单击 Ribbon 的开始选项页里的美元符号，如图 5-89 所示。选择当前的制造成本，如图 5-90 所示。



	A	B		C		D
1		Previous Year		Current Year		
2	Labor	$	16.00	$	16.80	
3	Benefits	$	6.40	$	9.25	
4	Materials	$	3.75	$	3.80	
5	Equipment	$	2.20	$	2.00	
6	Energy	$	1.25	$	1.45	
7	Overhead	$	14.00	$	[illegible]6.00	
8						

图 5-89

图 5-90

从 Ribbon 中选择【插入】|【饼图】命令，选择一个二维饼图，如图 5-91 所示，从设计组里选择一个【黑白】配色方案，如图 5-92 所示。

把生成的饼形图单独插入到另一个工作表里。

最后得到的结果如图 5-93 所示。

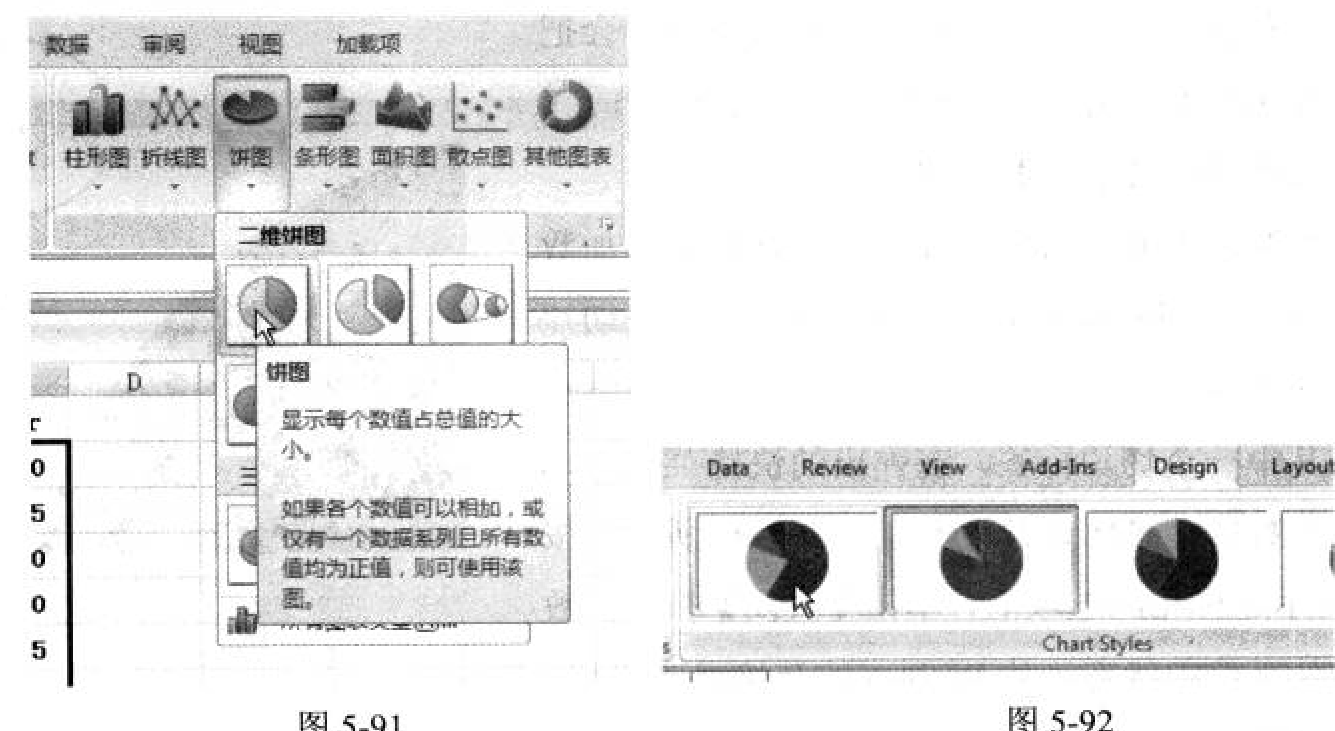


图 5-91　　　　图 5-92

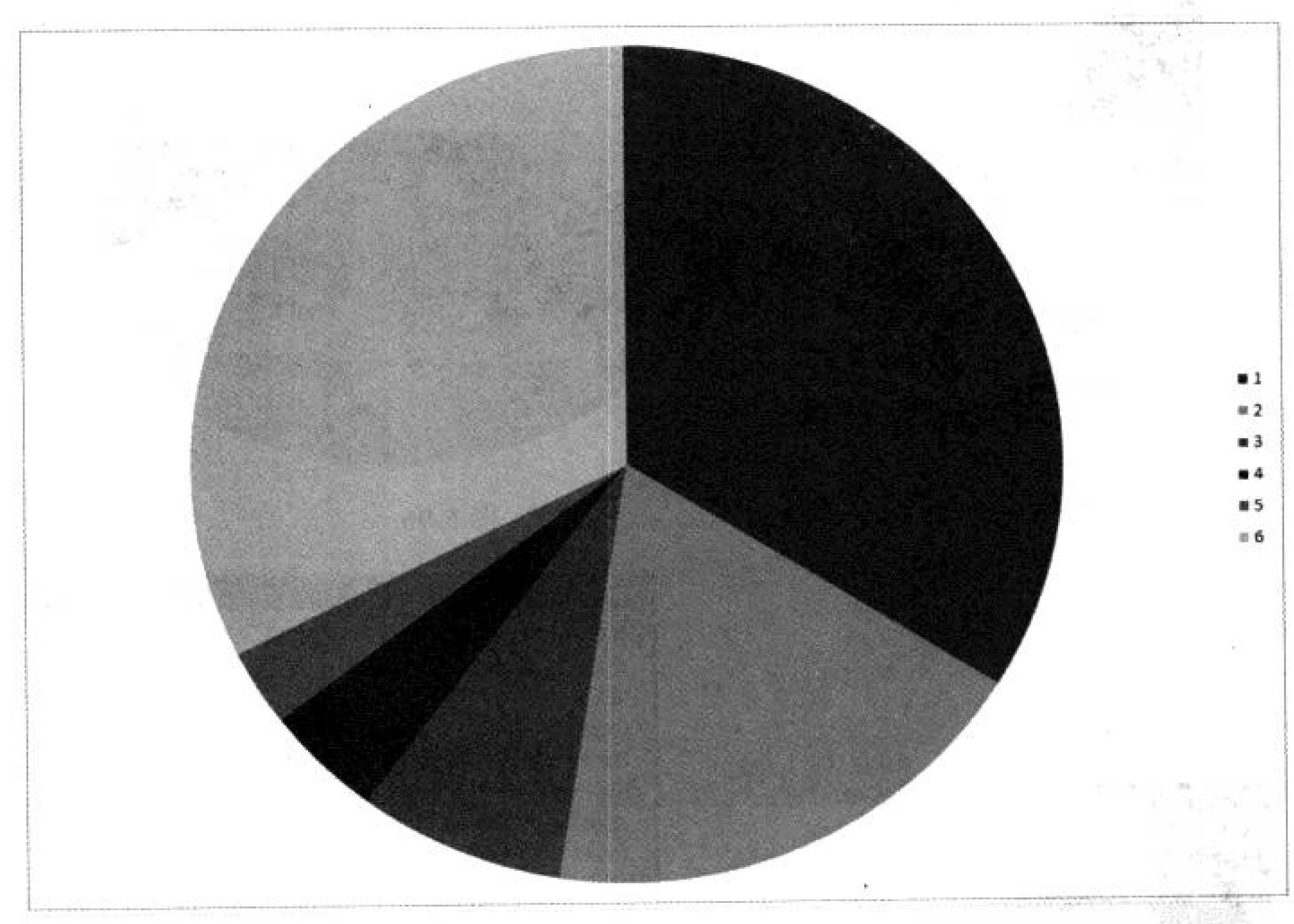


图 5-93

当饼图中的切片比较多时，即使使用带颜色的饼图，单凭图例理解饼图也是比较困难的。较好的办法是在每个切片的旁边加上相应的标签。

首先，要放置这些标签，必须缩小饼图。

单击紧靠饼图圆周的外部区域，选择饼图大小的调整框。单击并拖曳调整框的右上角，如图 5-94 所示，把饼图变小。此外，我们还可以在饼图的外部区域单击鼠标，把它拖曳到

图表的中央位置。注意，并拖曳饼图本身单击，会把饼图分解成为多个切片，假如读者不经意这样操作了，可单击工具栏上的【撤消】按钮。

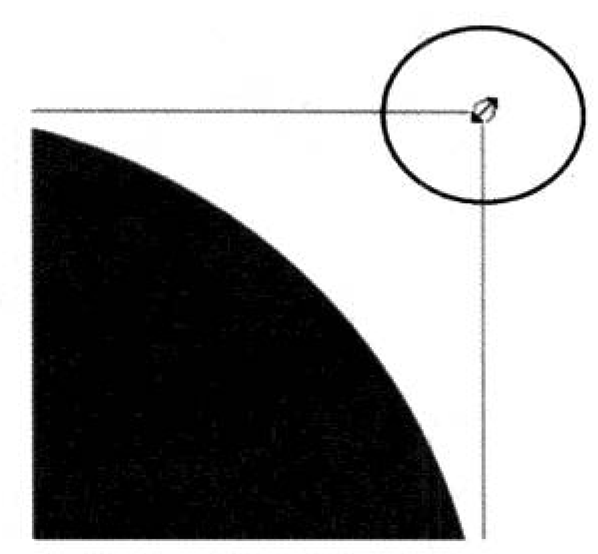


图 5-94

右击饼图，在弹出的快捷菜单中，选择【添加数据标签】命令，如图 5-95 所示。饼图将显示每个切片的值，如图 5-96 所示。

右击其中一个切片标签，在弹出的快捷菜单中，选择【设置数据标签格式】命令，如图 5-97 所示。按图 5-98 所示选择相应选项后，单击【关闭】按钮。数据标签还处于选中状态，从 Ribbon 的开始组选择较大的字体。

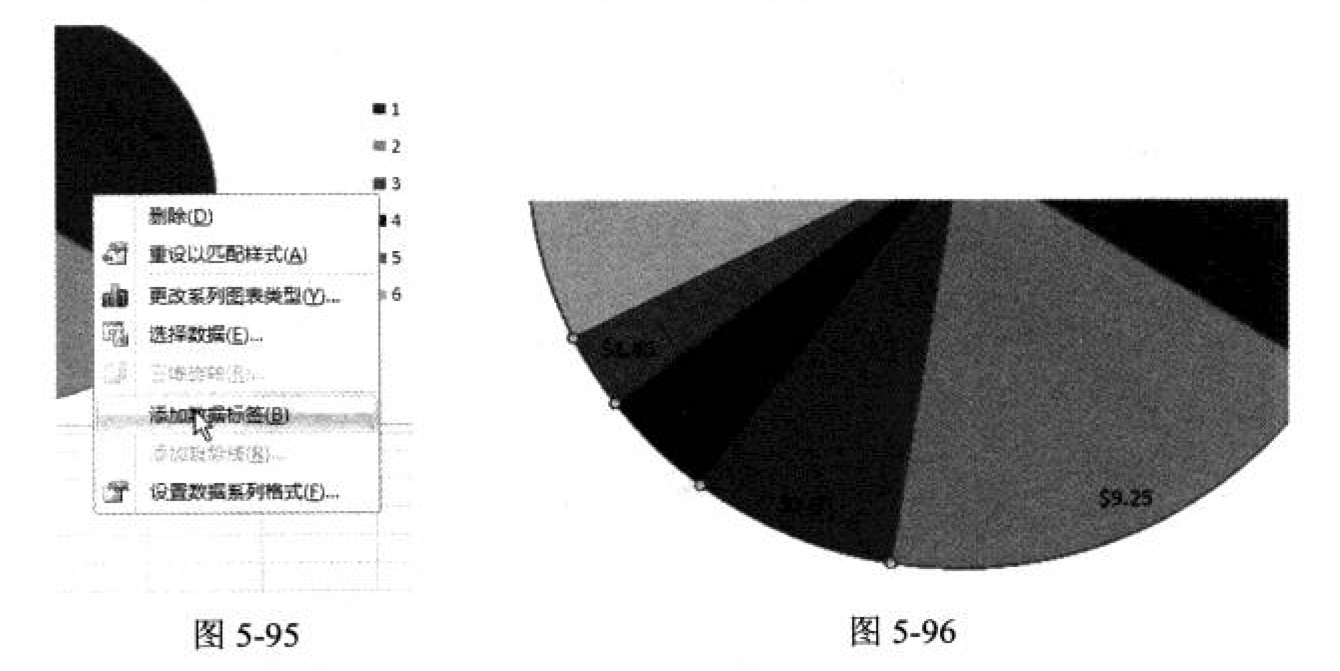


图 5-95

图 5-96

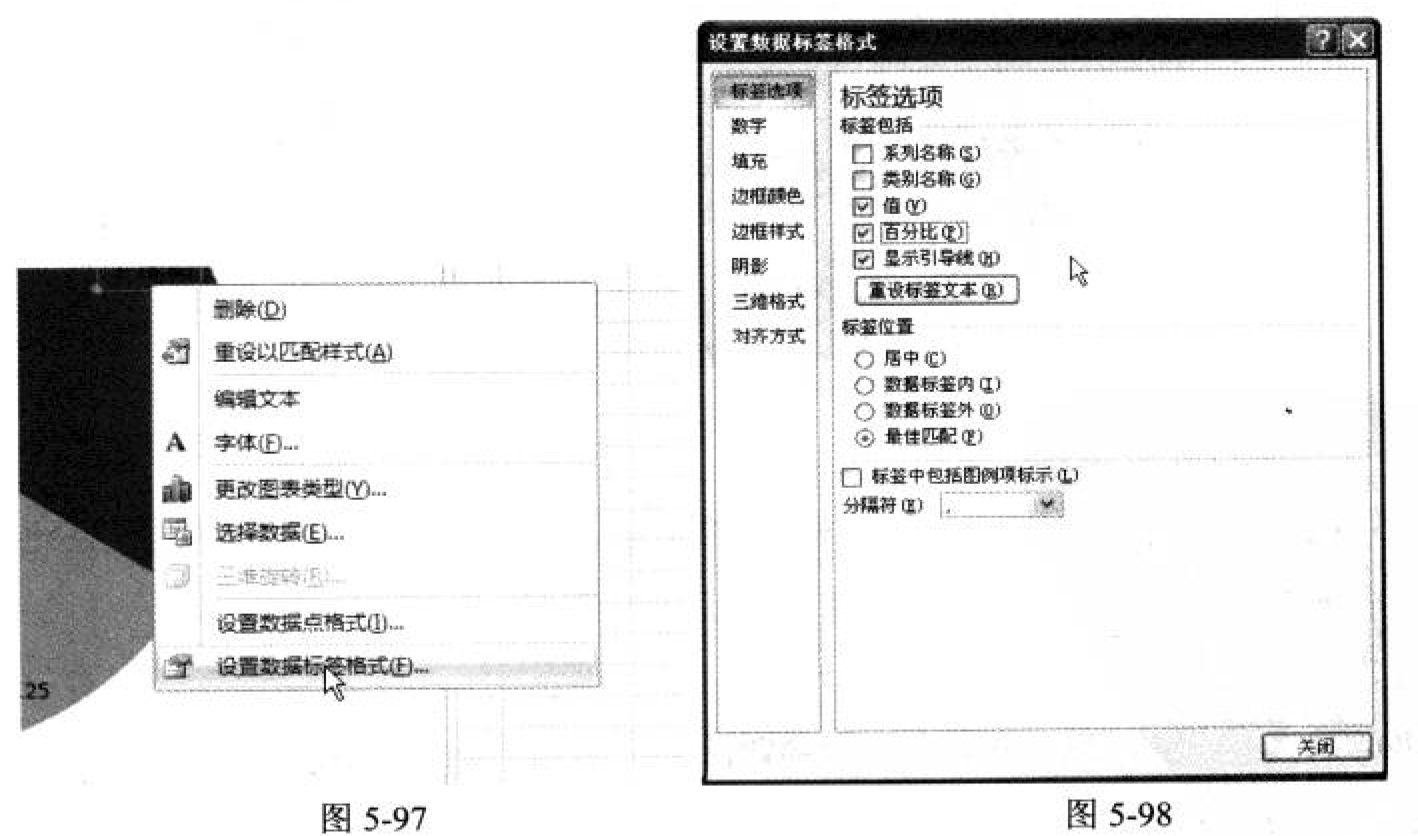


图 5-97

图 5-98

右击饼图，从弹出的快捷菜单中选择【选择数据】命令，如图 5-99 所示。

在弹出的【选择数据源】对话框中，单击【水平(分类)轴标签】下的编辑按钮，如图 5-100 所示，单击并拖曳到包含分类名字的单元格，如图 5-101 所示，单击【关闭】按钮。

单击其中一个数据标签，选择全部数据标签，再次单击其中一个标签，就会选择这个标签，取消选择其他标签。把这个标签拖曳到合适的位置，如图 5-102 所示。对其他标签重复上述步骤。

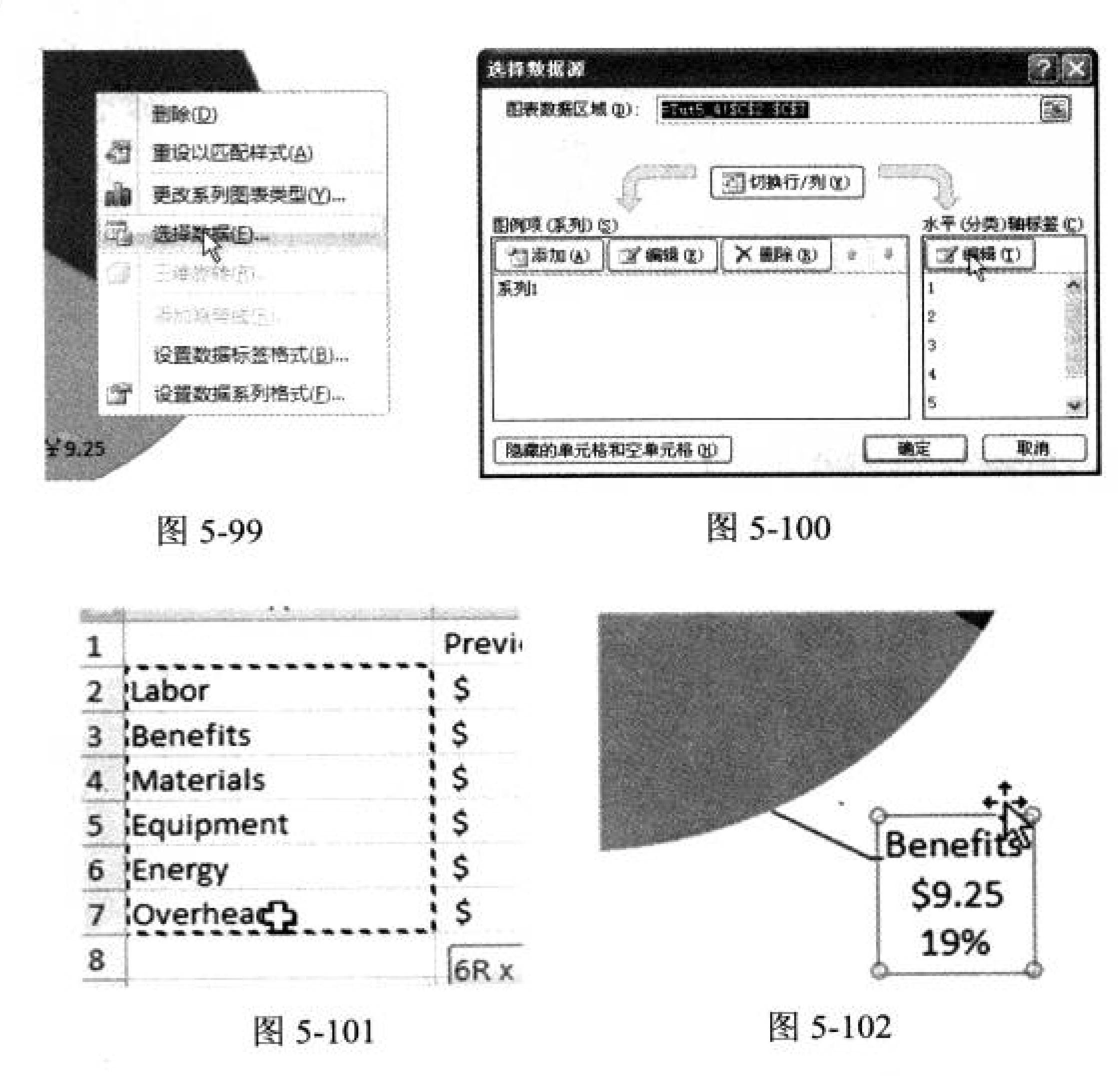


图 5-99

图 5-100

图 5-101

图 5-102

给每个切片增加边框。右击任意一个切片，在弹出的快捷菜单中选择【数据系列格式】命令，在弹出的【设置数据系列格式】对话框中选择【边框颜色】选项卡中的【实线】选项，选择黑色作为边框颜色，如图 5-103 所示。接着给图添加一个标题，然后选择图例，并按 Delete 键，删除图例。

最后得到如图 5-104 所示的图表。

还有其他选项可以修改图表的外观。其中之一是把饼图分解为各个切片，各个切片各自独立。

右击饼图，在弹出的快捷菜单中选择【数据系列格式】命令，把【点爆炸型】滑块向右移动，如图 5-105 所示，把滑块移动到大约 10%的位置后，单击【关闭】按钮。

得到的分散图如图 5-106 所示。分散饼图的另一种方法是，直接单击并拖动饼图本身。此外，有时为了突出某个切片，我们希望把其中一个切片与其他切片分离开。为此，应选择饼图，然后单击需要移动的切片，把它移动到另一个位置。

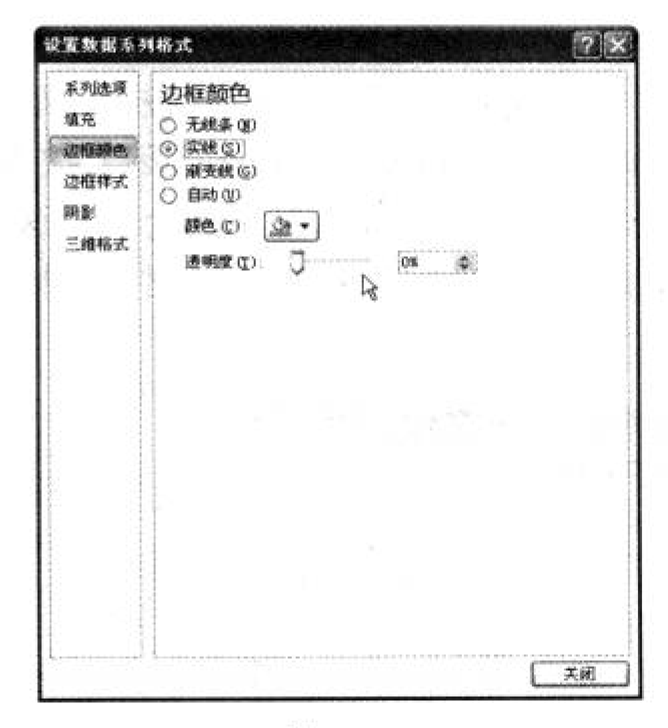


图 5-103

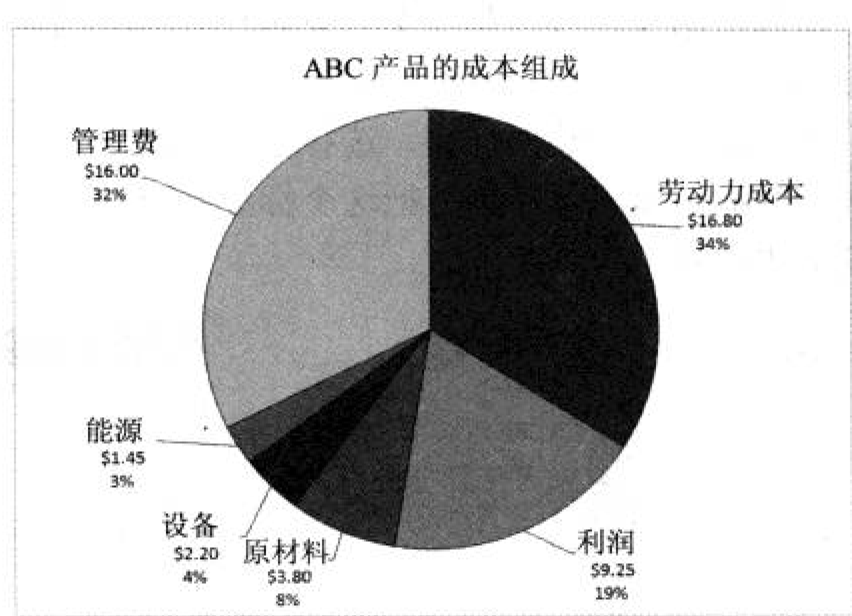


图 5-104

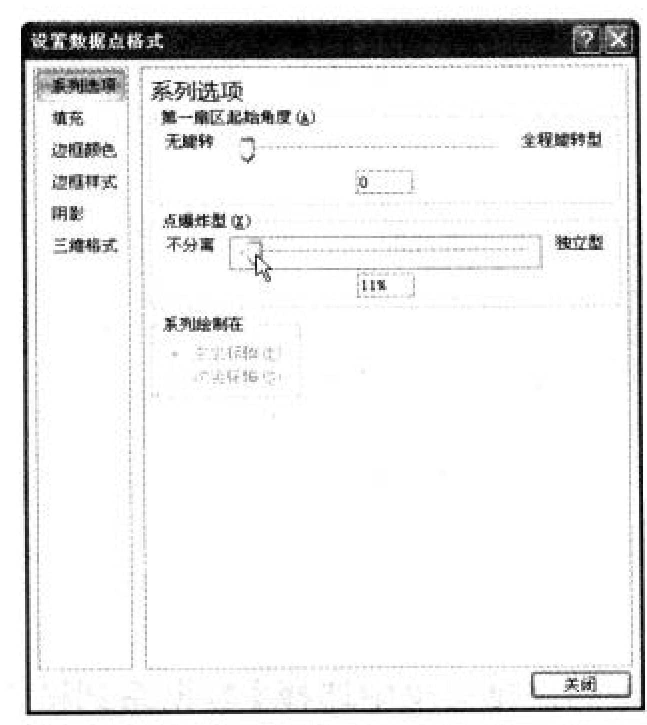




图 5-105

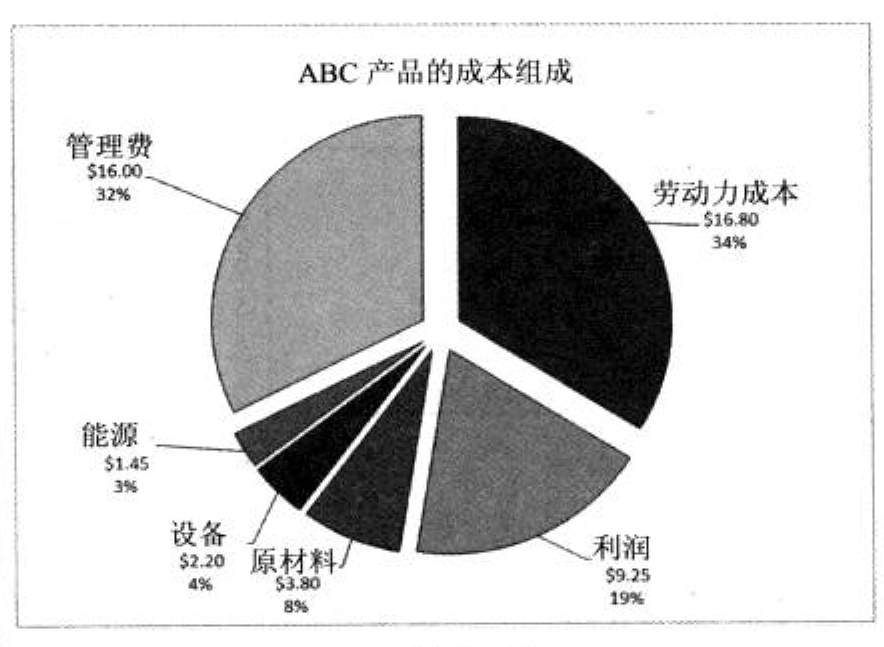


图 5-106

还可以将饼图设置成三维格式。

右击饼图周围的空白区域，在弹出的快捷菜单中选择【更改图表类型】命令，然后选择【三维饼图】命令，如图 5-107 所示。

右击此饼图，在弹出的快捷菜单中选择【三维旋转】命令，如图 5-108 所示。把绕 y 轴的旋转角度设置为另一个值，如图 5-109 所示。如果这个角度被设置为 90°，三维就变成为二维。这个角度越小，图表越平坦，试试各个不同的旋转角度，看看效果。

最后得到的三维分散饼图如图 5-110 所示。

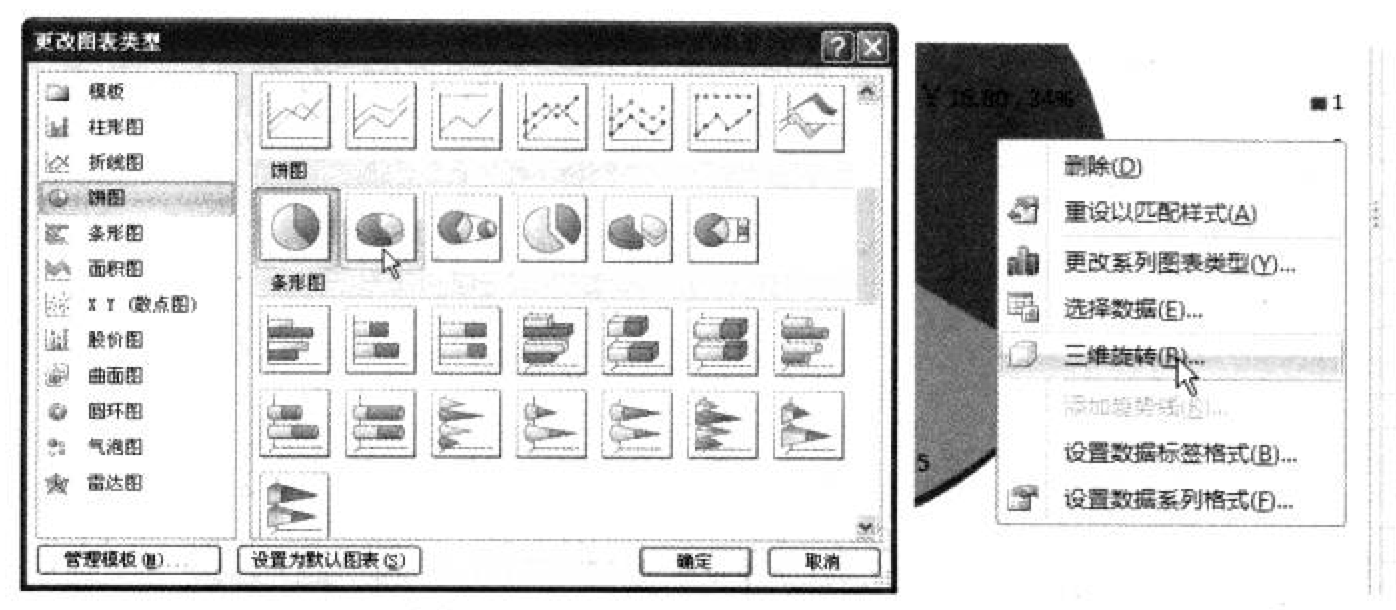


图 5-107　　　　　　　　图 5-108

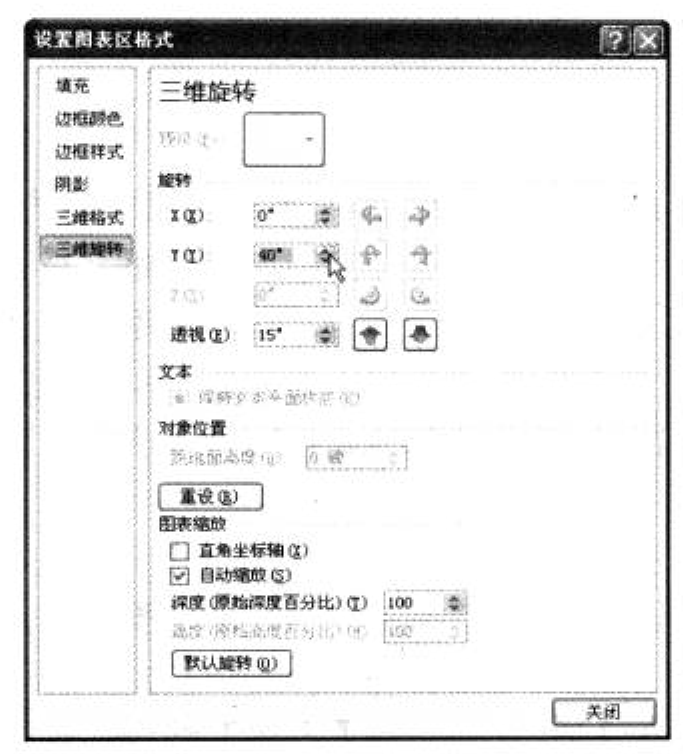


图 5-109

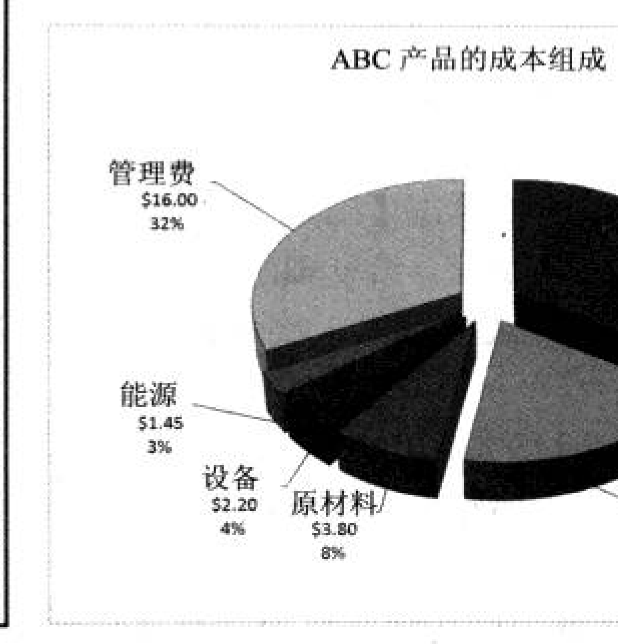


图 5-110

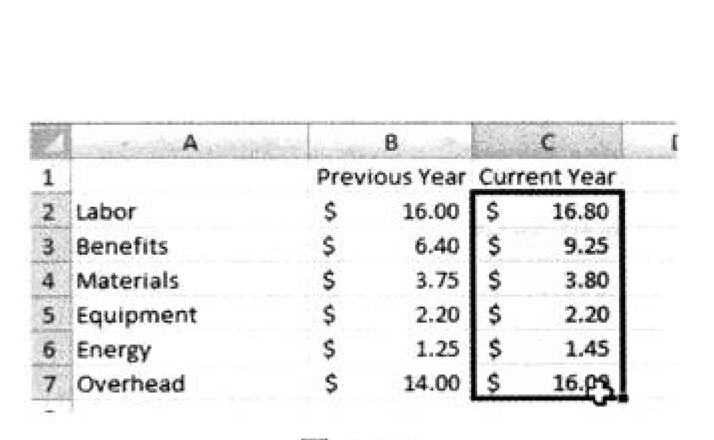
下面我们用柱形图显示这些数据。前面曾提到，柱形图是大家普遍接受的一个术语，不论柱形旋转方向如何。在 Excel 里，竖起放置的柱形图被称为条形图。

例 5.9

建立一个柱形图显示表 5-3 里前一年和当年的产品制造成本。

解：

选择当前成本数据，如图 5-111 所示。从 Ribbon 选择【插入 | 二维条形图 | 簇状条形图】命令，如图 5-112 所示。



	A	B	C
1		Previous Year	Current Year
2	Labor	$ 16.00	$ 16.80
3	Benefits	$ 6.40	$ 9.25
4	Materials	$ 3.75	$ 3.80
5	Equipment	$ 2.20	$ 2.20
6	Energy	$ 1.25	$ 1.45
7	Overhead	$ 14.00	$ 16.00

图 5-111

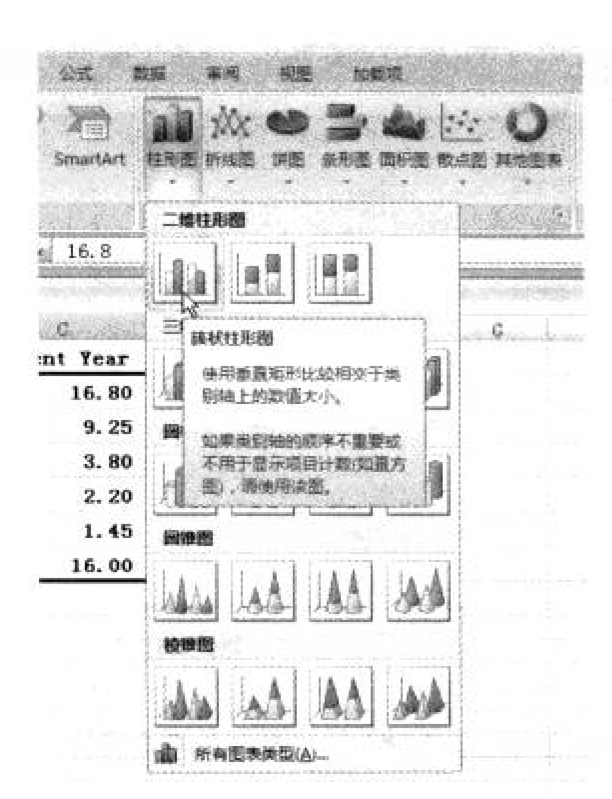


图 5-112

把图表移动到另外一个工作表里，右击此图表，在弹出的快捷菜单中选择【选择数据】命令，接着单击横坐标(分类)轴标签下面的【编辑】按钮，选择包含分类名称的单元格，单击【确定】按钮，再次单击【确定】按钮，关闭【选择数据】对话框。然后，添加一个标题，并对它进行格式设置。给 y 轴添加一个标签。右击图表，在弹出的快捷菜单中选择【设置图表区域格式 | 边框颜色 | 实线】命令，设置边框颜色为黑色，把网格线颜色设置为灰色。

最后得到如图 5-113 所示的图表。

如果图表只显示一列数据，则可以删除图例。但是，我们要添加前一年的数据，因此需要图例分区它们。

右击图表，在弹出的快捷菜单中选择【选择数据】命令，单击【添加】按钮，如图 5-114 所示。选择前一年的数据作为另一系列数据，如图 5-115 所示。

输入“Previous Year”作为该系列的名称，如图 5-116 所示。

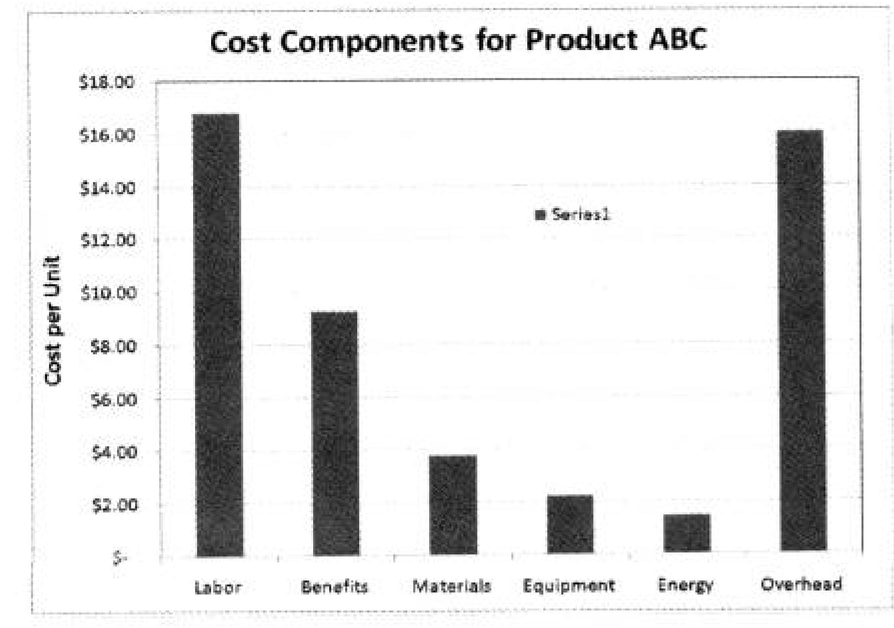


图 5-113

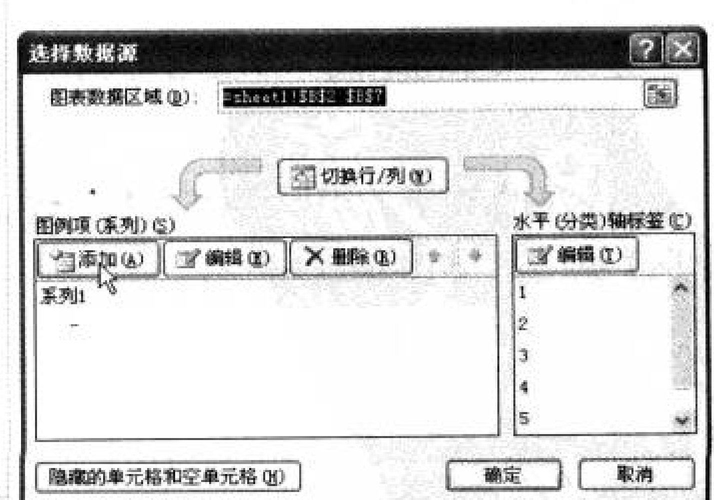


图 5-114

	A	B	C
1		Previous Year	Current Year
2	Labor	$ 16.00	$ 16.80
3	Benefits	$ 6.40	$ 9.25
4	Materials	$ 3.75	$ 3.80
5	Equipment	$ 2.20	$ 2.20
6	Energy	$ 1.25	$ 1.45
7	Overhead	$ 14.00	$ 16.00
8			6R x 1C
9		$ 43.60	$ 49.50

图 5-115

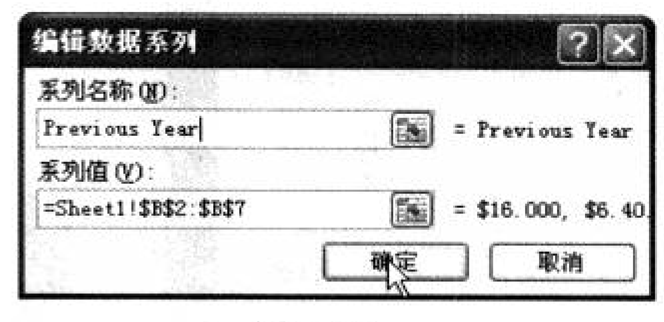


图 5-116

单击图表中“系列 1”的数据，选择【编辑】命令，把它重命名为“当前”。单击图 5-117 里的向下移动箭头，这样，前一年数据将显示在当前数据之前。

如果需要，我们还可以改变柱条的宽度。方法是：右击任意一个柱形条，在弹出的快捷菜单中选择【设置数据系列格式】命令，移动滑动条上的滑块，调整柱形条之间的重叠/间隔程度，如图 5-118 所示。上面的滑动条控制柱形之间的间隔或重叠程度，下面的滑动条控制各类之间的间隔。

最后得到的图表如图 5-119 所示。

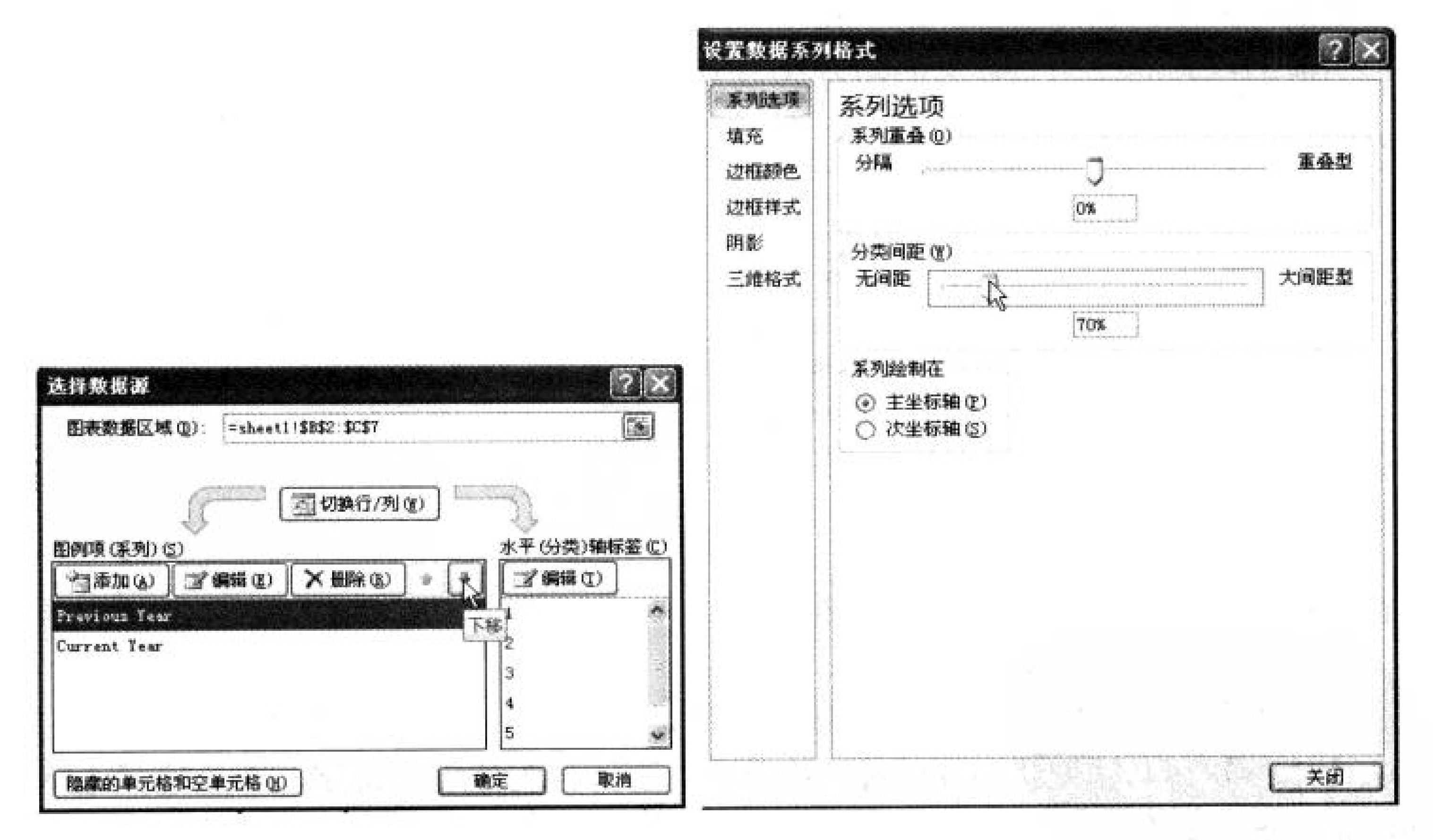


图 5-117　　　　图 5-118

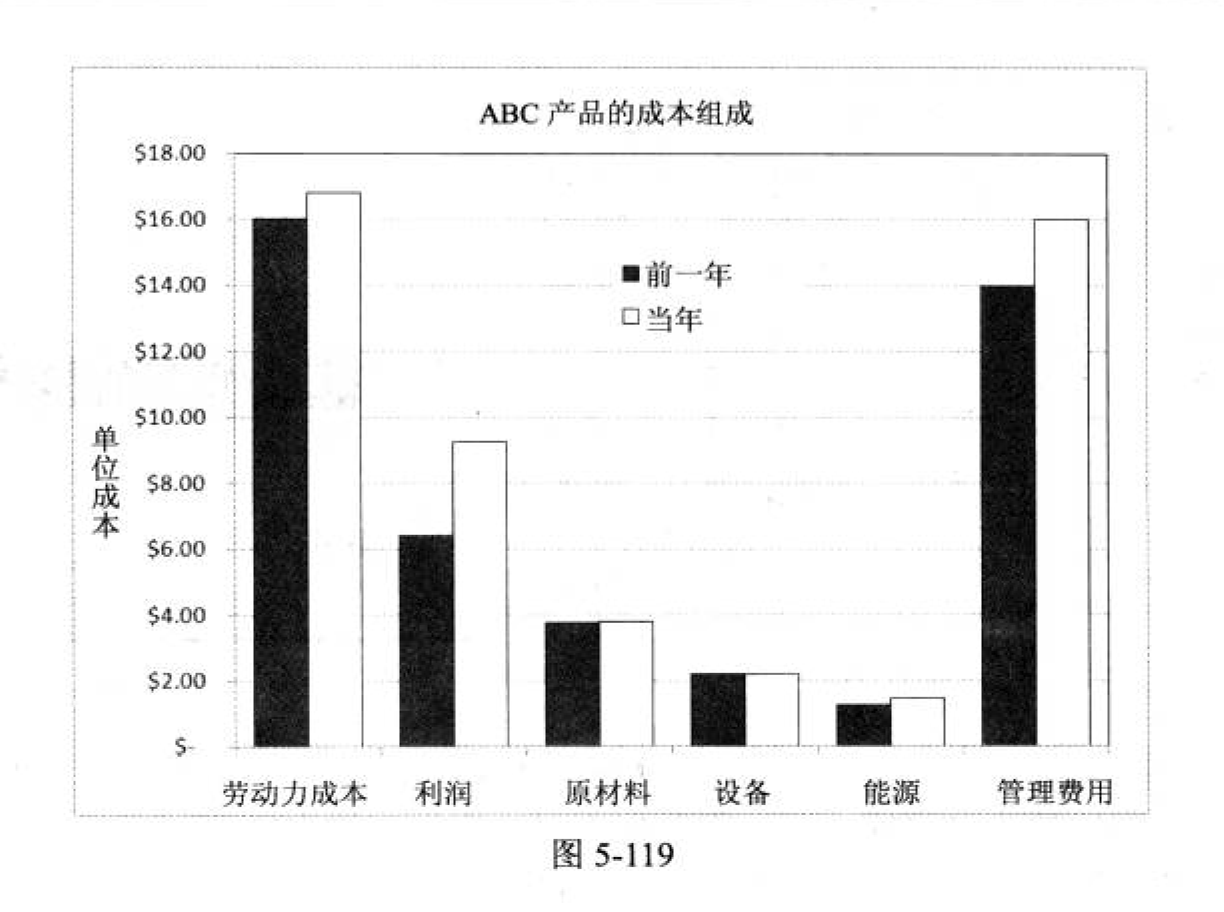


图 5-119

例 5.10

建立一个 Pareto 图，显示表 5-3 所示的当年制造成本。

解：

Pareto 图结合了柱形图和曲线图。即使读者从未使用过 Pareto 图，但通过这个例子读者也会掌握如何把两类图形结合在一起，如何使用两个 y 坐标轴。

把当前的数据和分类名称复制到另外一个区域，如图 5-120 所示。

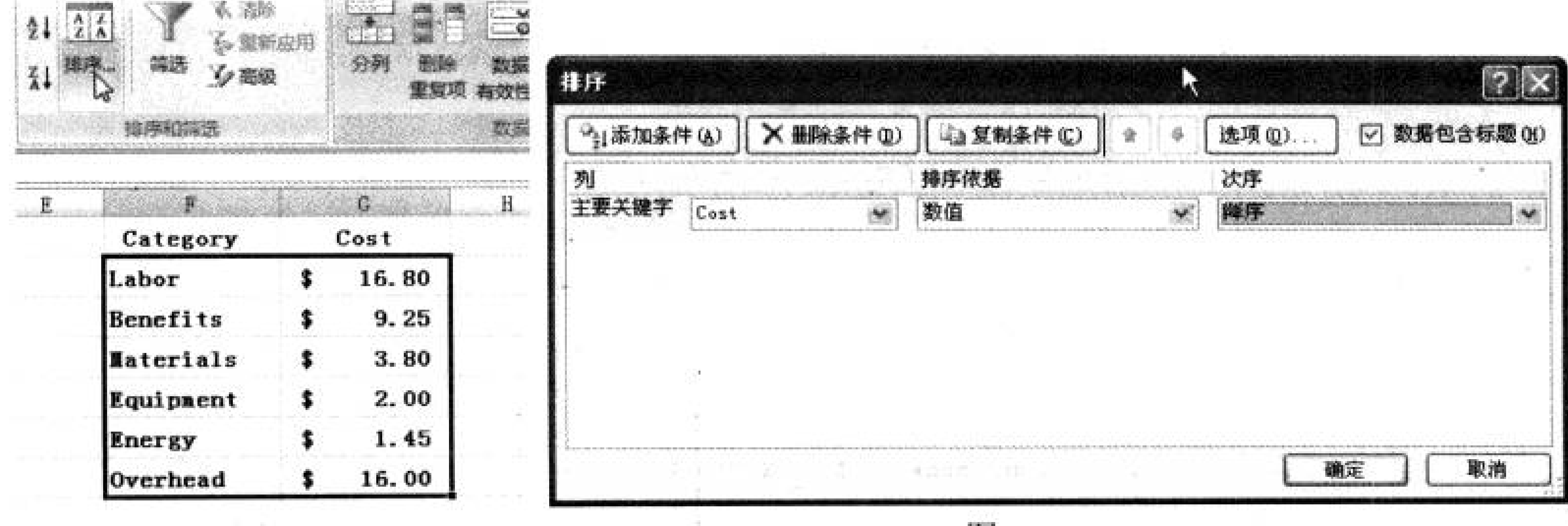
F	G
Category	Cost
Labor	$ 16.80
Benefits	$ 9.25
Materials	$ 3.80
Equipment	$ 2.20
Energy	$ 1.45
Overhead	$ 16.00

图 5-120

在 Pareto 图里，分类名称需要根据数值从高到低排序。因此我们必须先对数据进行排序。

选择分类名称和相应的数值。从 Ribbon 的数据组里选择【排序】命令，如图 5-121 所示，在【排序】对话框里，必须选择“Cost”的那一列为排序关键字，并且选择【降序】选项，如图 5-122 所示，取消选择【数据包含标题】选项。完成之后，单击【确定】按钮。

排序结果如图 5-123 所示。



Category	Cost
Labor	$ 16.80
Benefits	$ 9.25
Materials	$ 3.80
Equipment	$ 2.00
Energy	$ 1.45
Overhead	$ 16.00

图 5-121　　　　图 5-122

Category	Cost
Labor	$ 16.80
Overhead	$ 16.00
Benefits	$ 9.25
Materials	$ 3.80
Equipment	$ 2.20
Energy	$ 1.45

图 5-123

选择 Cost 一列数据下面的一个单元，输入公式计算成本之和。在劳动力成本右侧一个单元里(即图中的 H 列)，输入一个公式，把劳动力成本除以成本总和，如图 5-124 所示。

在 Overhead 右边的一个单元里，输入一个公式，将把管理成本除以成本总和，再加上一个单元格里的值，如图 5-125 所示。成本总和单元的地址必须使用绝对引用，这样复制公式时，这个地址不会改变。

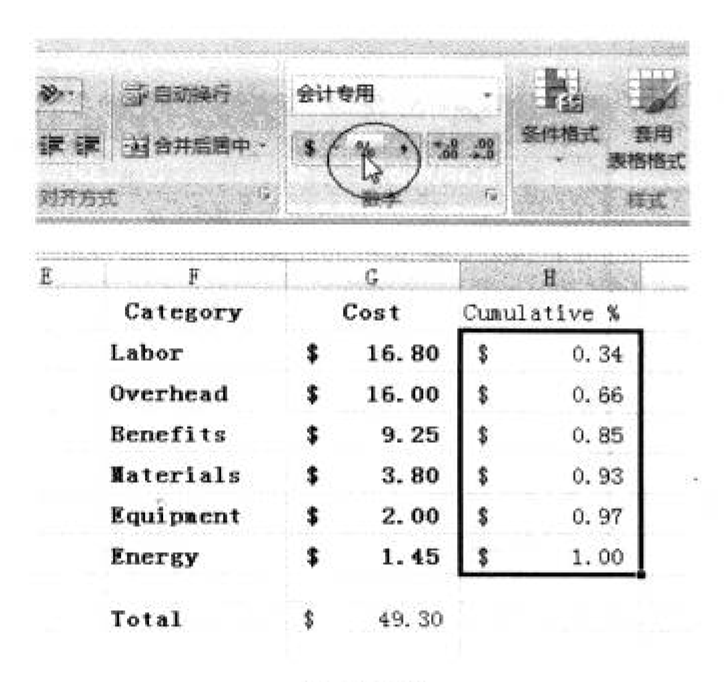
F	G	H
Category	Cost	
Labor	$ 16.80	=G2/G9
Overhead	$ 16.00	
Benefits	$ 9.25	
Materials	$ 3.80	
Equipment	$ 2.20	
Energy	$ 1.45	
Total	$ 49.50	

图 5-124

F	G	H
Category	Cost	
Labor	$ 16.80	$ 0.34
Overhead	$ 16.00	=G3/$G$9+H2
Benefits	$ 9.25	
Materials	$ 3.80	
Equipment	$ 2.20	
Energy	$ 1.45	
Total	$ 49.50	

图 5-125

把刚输入的公式复制到下面的单元格里。需要说明的是，按照默认设置，数据的显示格式为【会计专用】格式。选择这列的数据，并单击 Ribbon 开始组里的“%”格式，如图 5-126 所示。



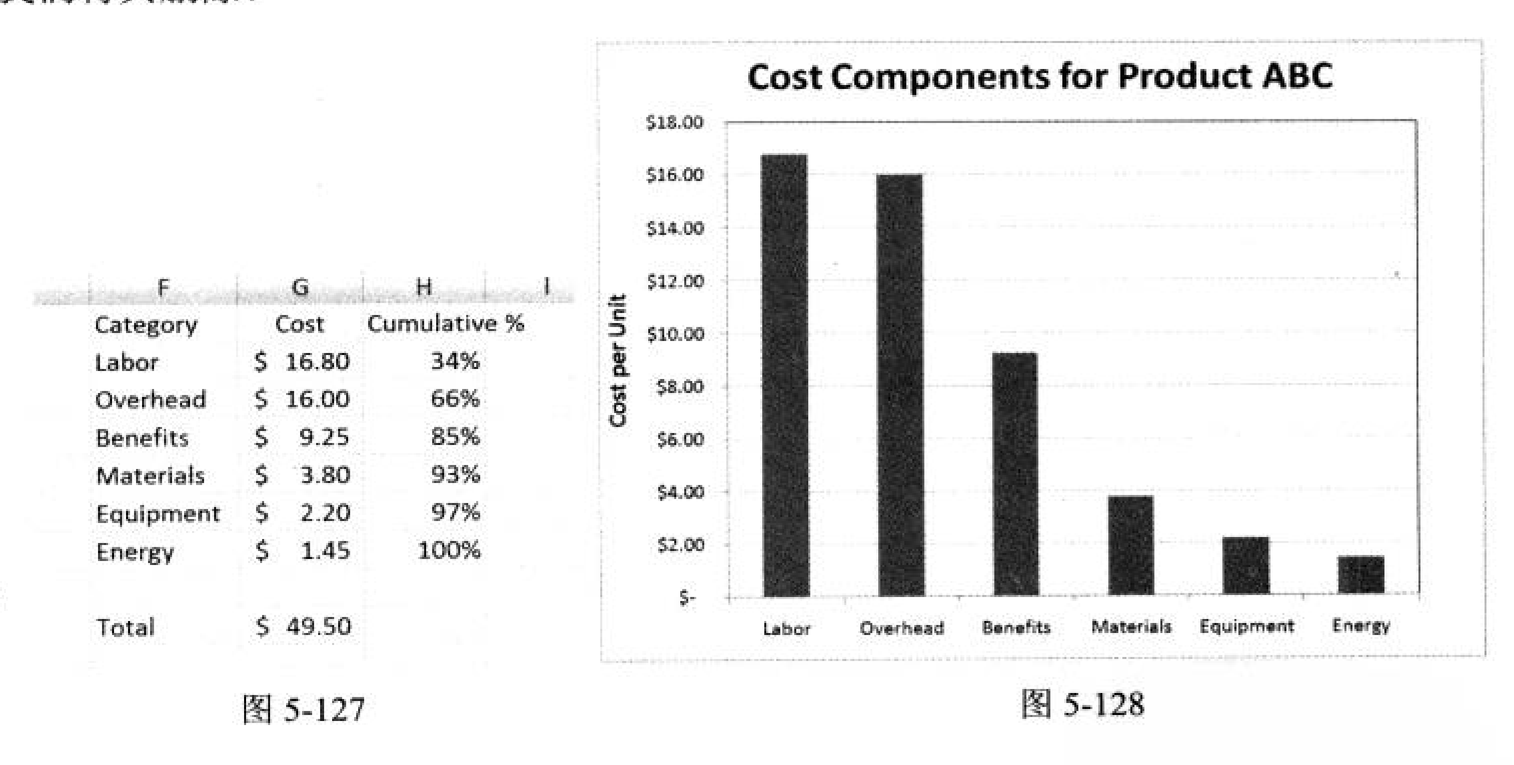
Category	Cost	Cumulative %
Labor	$ 16.80	$ 0.34
Overhead	$ 16.00	$ 0.66
Benefits	$ 9.25	$ 0.85
Materials	$ 3.80	$ 0.93
Equipment	$ 2.00	$ 0.97
Energy	$ 1.45	$ 1.00
Total	$ 49.30	

图 5-126

刚才建立的这一列表示了每一类成本的累加和。例如，第一个单元表示劳动力成本占总成本的 34%，第二个单元格的值表示劳动力成本与管理成本之和占总成本的 66%，其他单元类似。

把这一列设置为百分比格式，如图 5-127 所示。

建立一个新的柱形图表示这个累计成本比例，如图 5-128 所示。因为不需要图例，因此我们将其删除。



Category	Cost	Cumulative %
Labor	$ 16.80	34%
Overhead	$ 16.00	66%
Benefits	$ 9.25	85%
Materials	$ 3.80	93%
Equipment	$ 2.20	97%
Energy	$ 1.45	100%
Total	$ 49.50	

图 5-127　　图 5-128

在图形区域内部任意位置，单击鼠标右键，在弹出的快捷菜单中选择【选择数据】命令，单击【添加】按钮，新增一个数据系列，选择累加百分比作为此系列数据，如图 5-129 所示。也不需要给这个系列数据一个新的名称。

新的系列将以柱形条的形式出现在图表里。

右击其中任意一个百分比数据的柱形条，在弹出的快捷菜单中选择【更改系列图表类

型】命令，如图 5-130 所示。

注意，当我们在图表的空白区域单击鼠标时，就会出现一个选项，允许我们修改图表的类型。我们曾利用这个选项，把二维饼图改为三维饼图。当图中有多个数据系列时，单击其中的一个系列，也会出现这一个选项，通过它可改变这个系列的图表格式。

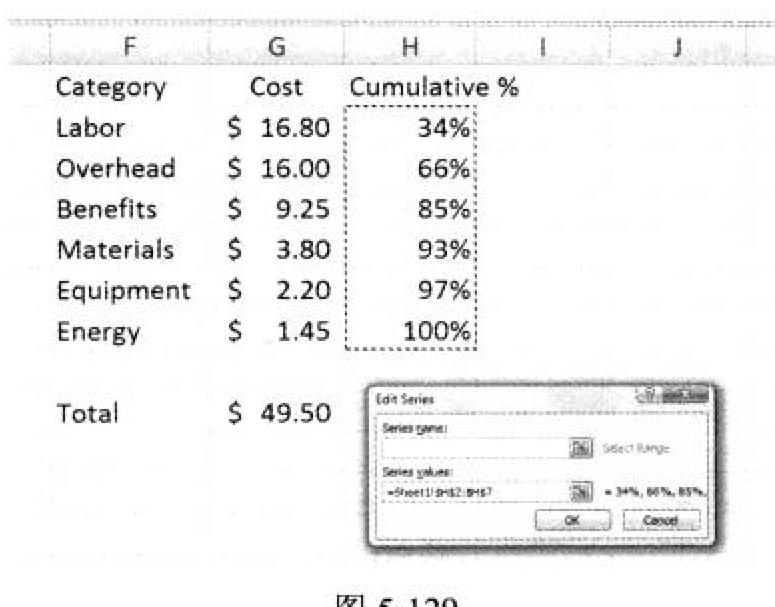


图 5-129

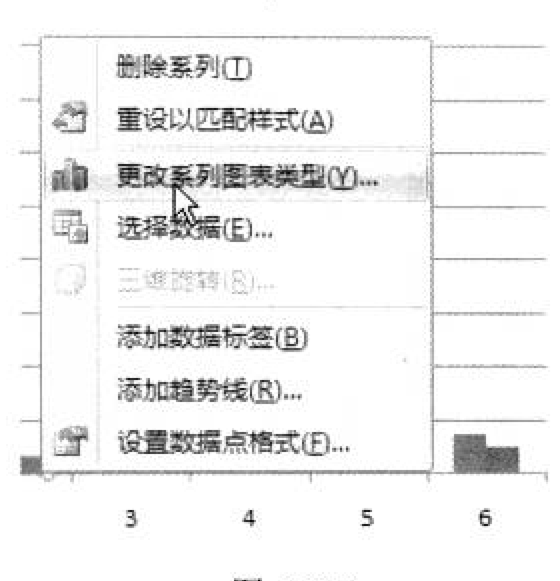


图 5-130

选择【带数据标志的曲线图】，如图 5-131。

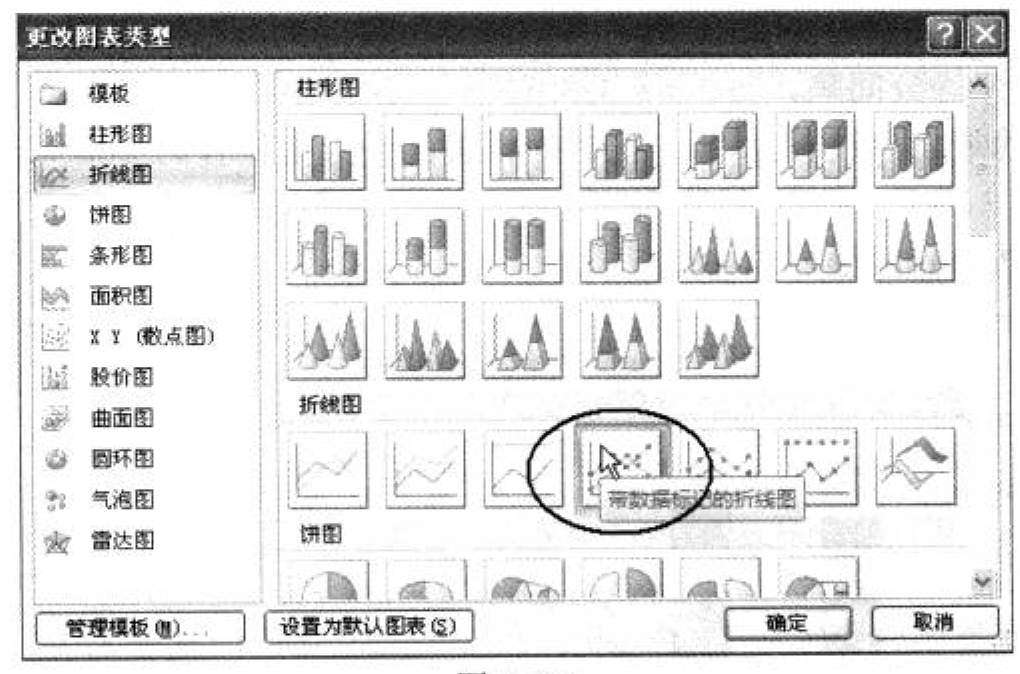


图 5-131

该曲线非常接近于横坐标轴，因为它是根据 y 轴的制造成本值显示的。为了使这个曲线能够显示在整个垂直范围内，我们必须用第二个坐标轴显示这个累积百分比。

右击任意累积百分比的任意一个数据标志，在弹出的快捷菜单中选择【设置数据系列格式】命令，如图 5-132 所示。选择【第二个坐标】选项，如图 5-133 所示。

最后得到如图 5-134 所示的图表。

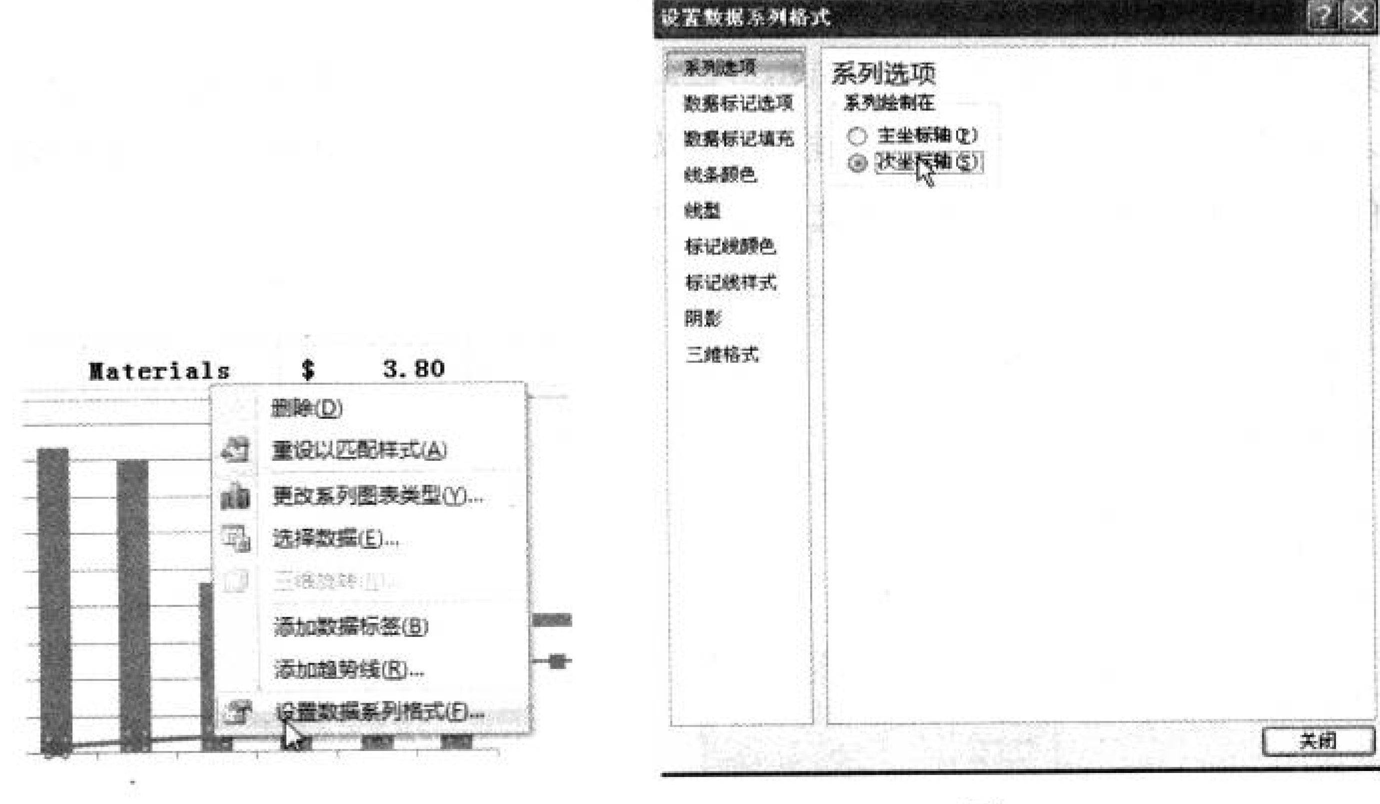


图 5-132 图 5-133

注意，图表中的网格线对应主坐标轴，与次坐标轴没有关系。为了使网格线同时能对齐到两个纵坐标轴，必须调整坐标轴的刻度值，使得相应等分间隔要相等。例如，我们把主坐标轴的刻度范围设置为 0~20 美元，并把次坐标轴的刻度范围设置为 0~100%，则两个坐标都会分成 10 个等分间隔。

把主坐标轴的最大值设置为 20，把次坐标轴的最大值设置为 100%，并给次坐标轴增加标签。

最后得到的 Pareto 图如图 5-135 所示。

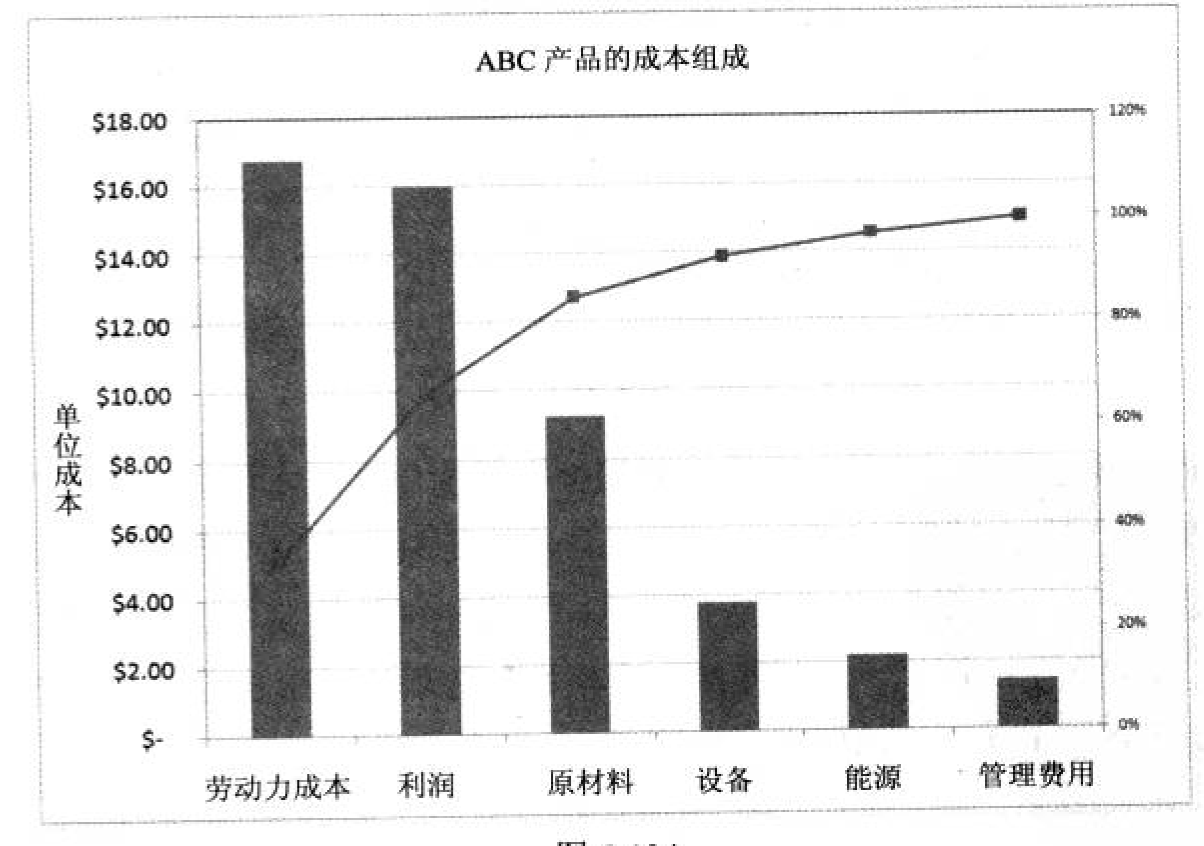


图 5-134

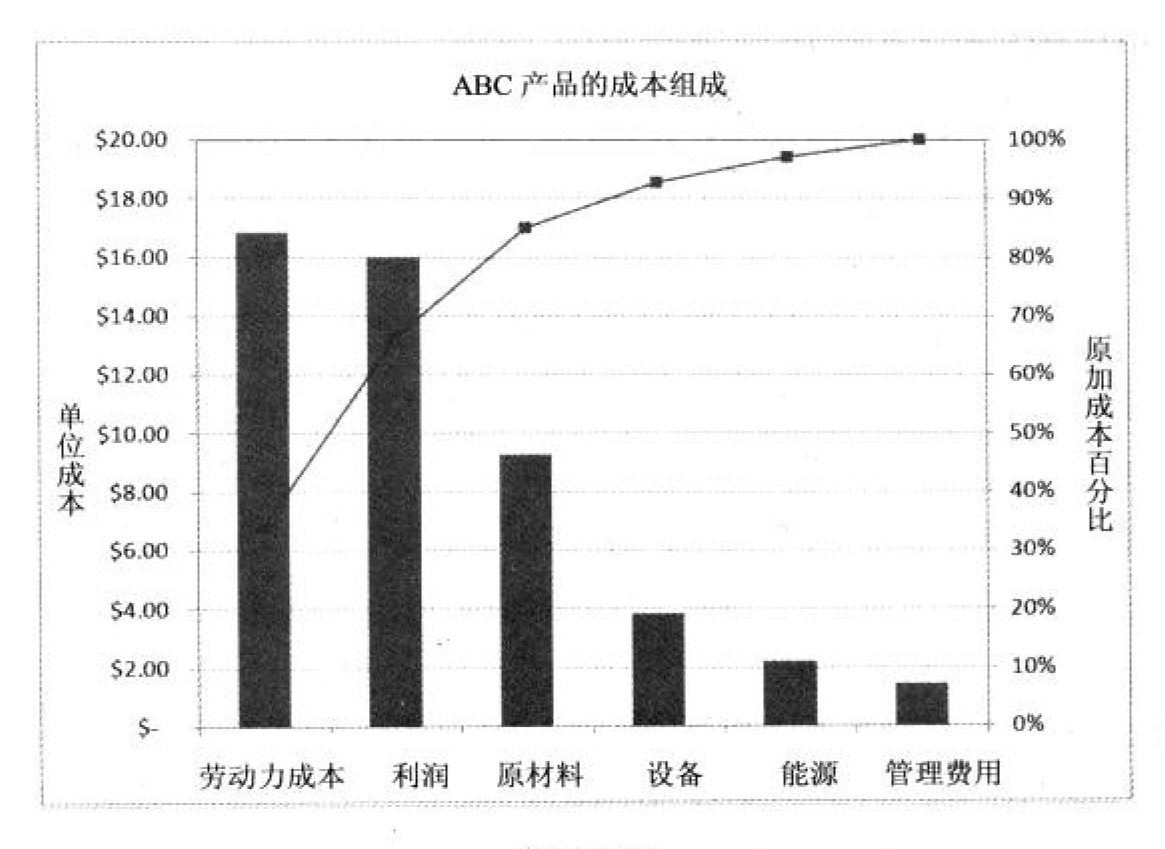


图 5-135

5.5　习题

1. 用 Excel 绘制以下方程的曲线，x 的取值范围为-3~6。

$$y = x^4 - 4x^3 - 6x^2 + 15$$

2. 用 Excel 绘制余弦曲线，角度取值范围为 0º~360º。
3. 用 Excel 绘制以下方程的曲线，x 取值范围为 0~10。

$$y = 3\sqrt{x}$$

4. 用 MATLAB 重做习题 1，要求用 for 循环和 plot 命令。
5. 用 MATLAB 重做习题 1，要求用 fplot 命令。
6. 用 MATLAB 重做习题 2，要求用 for 循环和 plot 命令。
7. 用 MATLAB 重做习题 2，要求用 fplot 命令。
8. 用 MATLAB 重做习题 3，要求用 for 循环和 plot 命令。
9. 用 MATLAB 重做习题 3，要求用 fplot 命令。

10. 在工程计算中经常用到双曲函数 cosh( )和 sihh( )。例如，一个绳子或一根缆线在自身的重力作用下，形成一个悬链线，悬链线的方程是由 cosh( )定义的。就像正弦函数和余弦函数与圆有关一样，这些函数与双曲线有关。Excel 和 MATLAB 都有内置的 cosh( )和 sinh( )函数。用 Excel 在同一个坐标轴上绘制 cosh( )和 sinh( )曲线，要求自变量取值范围为-2~2。并解释计算结果。

11. 在习题 10 的图里添加双曲正切函数 tanh( )曲线。

12. 用 MATLAB 重做习题 10 和习题 11。

13. 我们在第 2 章里曾遇到过轴承焠火问题。即在滚球轴承的制造过程中，各部件需要经过一道硬化工序。具体过程是先加热再迅速冷却，或者把它浸入到油槽或水槽里，这个过程即为淬火过程。滚球轴承的温度是时间的函数。$T(t)$可以按方程(2.3)估算：

$$T(t)=(T_i-T_\infty)e^{-\frac{t}{\tau}}+T_\infty$$

式中 t 是轴承浸入槽中的时间，单位为 s。T_i 是轴承的初始温度，T_∞ 是油的温度。τ 是时间常数，单位是 s，它与轴承的材料、轴承的几何形状及油的特性有关。要求建立一个电子表格，以 T_i、T_∞ 和 τ 为输入变量，计算在 0~180 之间，步长为 1s 的各时刻轴承的温度随时间的变化。假设时间常数 τ ＝50s。假设轴承的初始温度为 800ºC，油的温度为 40ºC。建立一个图表显示轴承的冷却过程。

14. 在图 5-13 生成的图表里添加两个曲线，分别表示时间常量为 20 和 100 的冷却曲线。延伸 3 个曲线的时间，直到温度为 80 度为止。

15. 用 MATLAB 重复习题 13 和习题 14。用 while 循环生成曲线需要的数组，这 3 个曲线的循环的终止条件都是温度小于 80ºC。建立第二个图表，延伸时间，值到温度为 60ºC 为止。

16. 绳子通过一个轮子升降一个物体，如图 5-136 所示。由于绳子与轮子之间存在摩擦力，因此，绳子一端的受力要大于另一端的受力。在本例里，需要的力大于 500N 才能拉起一个重达 500N 的物体，这是因为在升起过程中存在摩擦力的缘故。在静力学中，读者知道，这两个力之间的关系为：

$$T_2=T_1e^{\mu\theta}$$

式中，μ 是摩擦系数，反映了两个相互接触物体之间的性质。θ 是两个接触点形成的角度，单位为 rad。

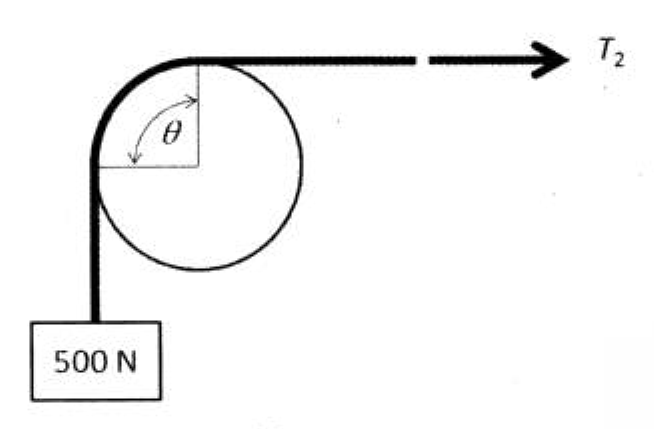


图 5-136

用 Excel 绘制拉起物体所需要的力，即公式中的 T_2 与角度的关系。角度取值范围为 0ºC~720ºC (720ºC 表示绳子绕轮子两圈)。假设 T_1=500N。并解释计算结果。

单击 y 坐标轴，选择【设置坐标轴格式】命令，把它设置为对数格式，并把坐标轴标签改为“对数值”(省略以 10 为底这几个字)。按这种格式绘制得到的曲线会是什么形状？我们称这样的曲线为半对数曲线，因为只有一个坐标轴使用对数刻度。

17. 考虑图 5-137 的四分之一圆。对于每个图，我们根据四分之一圆所包围的格子数，求它的面积。我们只把那些有二分之一或多于二分之一部分位于圆内的格子考虑在内。或者为了求得更准确的值，估算格子面积中圆所占的比例。用 Excel 绘制圆在格子里的面积与半径的关系，并添加趋势曲线拟合这些数据点。比较通过趋势线得到的方程与四分之一圆面积的精确计算公式。如果把图中的 x 和 y 轴都采用对数刻度，则将得到什么形状的曲线？

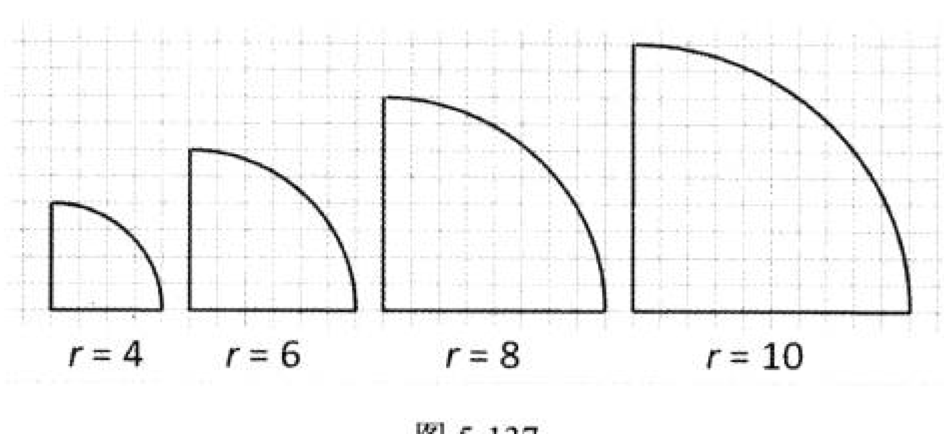


图 5-137

18. 表 5-4 是 30 个弹子球的直径。用 Excel 建立一个直方图，显示弹子球的直径分布。

表 5-4 弹子球的直径(单位为 in)

0.476	0.471	0.484	0.477	0.486	0.488
0.481	0.462	0.484	0.489	0.451	0.478
0.469	0.479	0.452	0.484	0.488	0.485
0.482	0.474	0.461	0.478	0.473	0.475
0.469	0.467	0.484	0.463	0.475	0.489

特别需要注意的是，当我们选择箱体时，最好选择 5~15 个箱体。箱体的取值间隔为 0.0025in 或 0.005in。读者可以用手工，或者用 if 语句，或者用数据分析插件中的 Histogram( ) 函数求得各个箱体的数据点数。

第 II 部分

工程计算的应用



第 6 章

求方程的根

引言

计算工具的一个常见应用就是求解代数方程。MATLAB 和 Excel 都包含强大的工具用来求代数方程的根。本章将讨论各种类型的代数方程，并且介绍如何选择合适的求解工具。

本章，我们将学习以下内容：

- 区别线性方程、多项式方程和普通非线性代数方程。
- 用二分法和牛顿迭代法求方程的根。
- 了解插入法和开型法的重要区别。
- MATLAB 里 polyval、roots、fplot 和 fzero 的用法。
- Excel 的单变量求解(Seek Goal)。
- 能够根据不同类型的方程选择合适的计算工具。

6.1 学习本章的目的

假设我们要为建筑工程单位设计一个横梁，横梁的结构如图 6-1 所示。

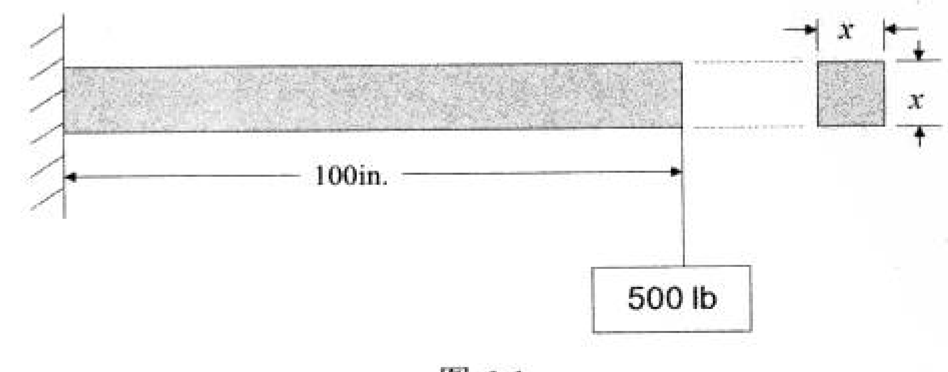


图 6-1

我们要决定这个钢梁的截面积的大小，要求它的强度足够大，可以支持在另一端悬挂

一个 500lb 重的物体。另一方面，为了节省横梁的钢材用料，我们只希望这个钢梁在临界点时，它的张力正好等于它受到的作用力。从材料力学可知，根据图 6-1 的尺寸，横梁受到的最大拉力由方程(6-1)决定：

$$\sigma = \frac{300\ 000}{x^3} + \frac{85\ 800}{x} \tag{6-1}$$

式中，σ 是每平方 in 的张力，x 是横截面的边长，单位为 in。方程(6-1)考虑了物体重 500lb 和横梁自身的重量。

利用材料力学方面的知识，我们知道，钢梁的每平方 in 可允许的张力为 12 000lb。因此要根据下面的方程确定 x 的值：

$$\frac{300\ 000}{x^3} + \frac{85\ 800}{x} = 12,000 \tag{6-2}$$

根据方程(6-2)，就可以求得能够支持 500lb 物体的最小横截面积。为了选择合适的钢梁，必须求解方程(6-2)。

在这个例子里，我们利用工程分析方法，把一个设计问题简化为一个代数方程。但是从工程实际问题中得到的方程往往非常复杂，经常不能简单地求得它们的解。本章，我们将学习如何用数值方法解这些复杂的代数方程。

6.2 方程求根：理论

考虑如式(6-3)所示的代数方程：

$$x^3 + 6x = \sin(2x) \tag{6-3}$$

这类方程的求解过程，就是寻找 x 的一个值以满足上述表达式。为了把上述方程当作一个经典的求根问题，我们首先对上述方程进行变换，把式中的全部项移到等号的一侧。如下所示：

$$f(x) = x^3 + 6x - \sin(2x) \tag{6-4}$$

现在方程(6-3)的求根问题就变成，如何寻找 x 的值，满足：

$$f(x) = 0 \tag{6-5}$$

任何满足上述方程的 x 值，都被称为方程的根。针对不同方程，方程可能不会有根，可能只有一个根，或有任意个根(无限个)。

6.2.1 方程的分类

对于一些特别简单的方程，以及其他一些特殊的情形，可以求得它们的解析解。但很多时候，我们不得不借助于数值算法求解代数方程。MATLAB 和 Excel 都提供了许多功能强大的代数方程求解工具。至于具体使用哪一个工具，就要取决于方程的类型。因此，我们先讨论代数方程的分类。

线性方程和非线性方程

通常我们把代数方程分成两大类：线性代数方程和非线性代数方程。

所谓的线性是，是指未知量 x 在方程中的指数只是 1。例如下面这 3 个方程都是线性代数方程：

$$7y-6=0 \tag{6-6}$$

$$\frac{z}{3}=-8 \tag{6-7}$$

$$8x-3=5x+4 \tag{6-8}$$

上述方程都可以转换成像 f(x)=Ax+B＝0 式子的标准形式，如下所示：

$$f(y)=7y-6=0 \tag{6-9}$$

$$f(z)=z+24=0 \tag{6-10}$$

$$f(x)=3x-7=0 \tag{6-11}$$

线性代数方程出现在绝大多数导论性质的代数课程里。它们只有一个根，而且用解析法可以准确求得它们的根。

其他任何通过简化不能成为线性方程或者不属于线性代数方程的方程都称为非线性代数方程。非线性方程含有自变量的指数次数不是 1 的项(如 x^2 或 $x^{1/2}$)和自变量的非线性函数(如 sin(x)和 log(x))。非线性代数方程可能没有根，可能只有一个根，或有无数个根。甚至有实根和虚根。除了少数简单的、特殊的例子外，大部分非线性方程的根都很难求得。事实上，很多非线性方程根本不存在解析解。

下面我们进一步对非线性代数方程进行分类。

多项式非线性方程和一般非线性方程

多项式方程是指方程经过多项移项和合并后，自变量的指数次数都是正整数(或者再乘上一个常量)或 0，例如：

$$f(y)=3y^7+5y^3-6y^2-12 \tag{6-12}$$

这就是多项式方程的一个例子。自变量 y 的指数只有正整数次。有时，方程需要化简后才能得到这种形式，分析下面的代数方程：

$$6y^2-\frac{3}{y}=11y+4 \tag{6-13}$$

初看起来，它不像是一个多项式方程。但是这个方程经过化简后的形式如下：

$$f(y)=6y^3-11y^2-4y-3=0 \tag{6-14}$$

多项式方程的次数是指把方程按标准形式排列后，自变量的最高指数。上述两个方程分别是 7 次和 3 次多项式方程。方程的次数之所以很重要，是因为，第 n 次多项式方程最多有 n 个实根。它可能有少于 n 个实根，其中一些根是虚根。在某些特殊情形，多项式方程的根可以用解析方法求得，例如，对如下形式的二次多项式方程：

$$f(x)=Ax^2+Bx+C \tag{6-15}$$

它的根可以由以下二次式公式精确求得：

$$y=\frac{-B\pm\sqrt{B^2-4AC}}{2A} \tag{6-16}$$

它最多有两个实根。然而，在一般情况，多项式方程的根需要使用数值求解法或作图法。

其他所有经过化简后不能变换或者不能归类为多项式方程的非线性方程统称为普通非线性方程。它们包括自变量的次数不为整数的方程(如 $x^{2/3}$)，或者包含自变量的非线性函数(如 sin(*x*)和 log(*x*))的方程。方程(6-4)就是一个普通非线性代数方程的例子。对于普通非线性方程，无法确定根的类型和根的个数。它可能没有根，或者一个根，或者无数个根。

我们用图表示方程的分类，如图 6-2 所示。

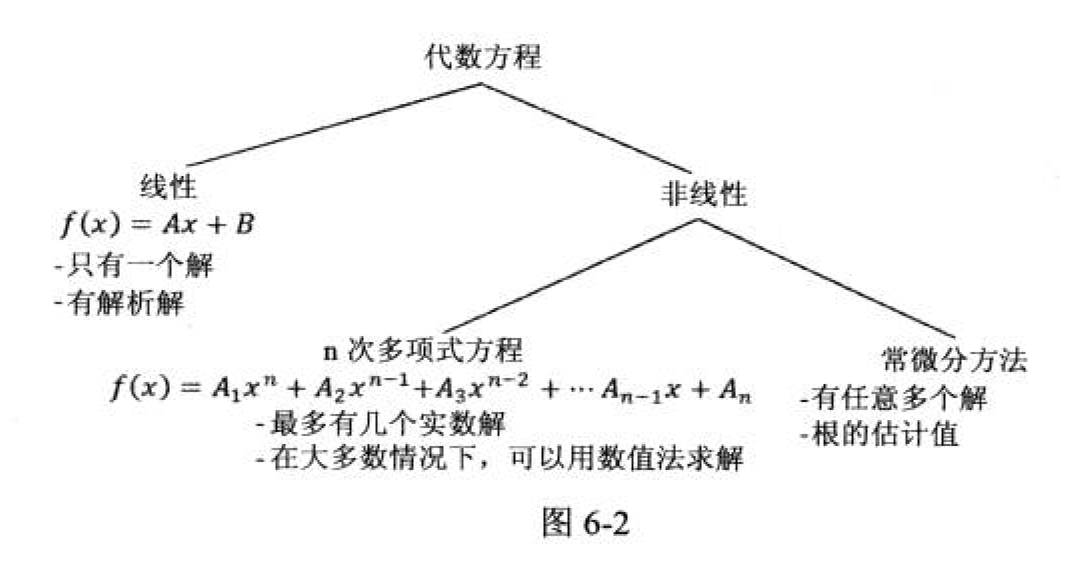


图 6-2

6.2.2　方程的解

当方程变成标准形式时，即所有项都移到方程的一侧，方程的求根问题可以看成是求满足以下公式的自变量 *x* 的值：

$$f(x)=0 \tag{6-17}$$

从技术上讲，我们根据方程的分类，确定方程求解的个体方法。本节介绍方程求根的基本理论。首先以一个非常简单的线性方程为例子。通过这个例子，介绍方程求根的几个基本概念，而同时我们尽量把数学问题简单化。

这个例子，要分析方程(6-9)描述的函数，为此，我们重写这个方程：

$$f(y)=7y-6 \tag{6-18}$$

假如要确定这个方程的根的位置，或者使 $f(y)=0$ 的 y 值。采用作图法可以在 *y* 的某个范围内，画出 *f*(*x*)的图形，读出 *f*(*y*)与水平轴相交点的位置，即相交位置的 *y* 值。如图 6-3 所示。

仔细分析这个图，我们发现，根据这个图的刻度，可以估计方程的根。方程的根在 0~1 之间。即在下限 $r_l=0$ 与上限 $r_u=1$ 之间的某个值。大致估计一下，把这两个点的中点作为根的位置，因此，我们说这个方程的根近似等于 *0.5*，如图 6-4 所示。

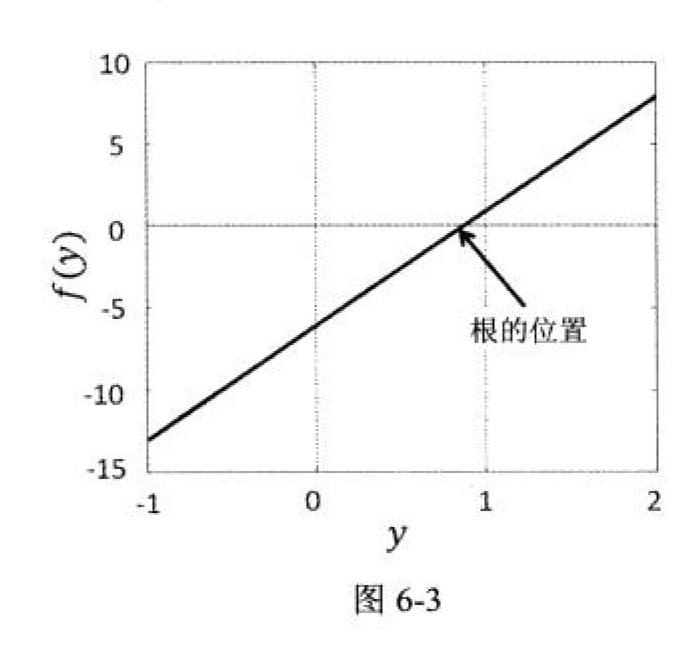


图 6-3

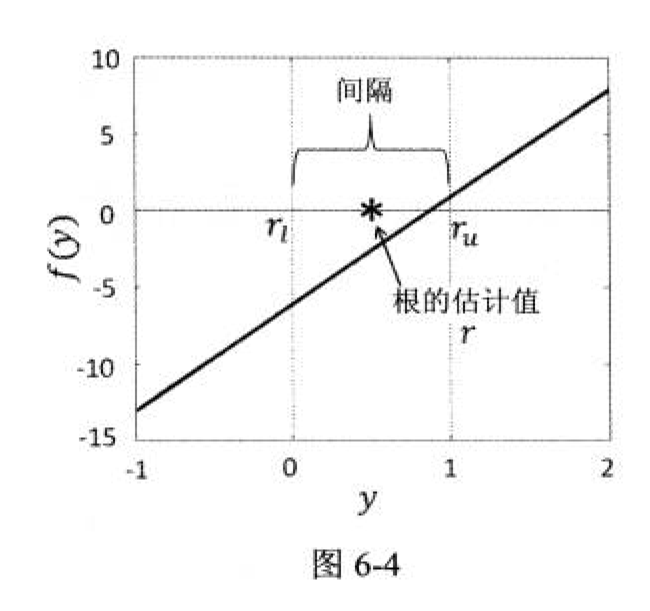


图 6-4

这个值算不算方程的准确解呢？这要看具体的应用。如果我们需要比较准确的结果，则可以用较小的刻度重新表示这个图，即让水平轴的间隔线更密集些。从图 6-5 的刻度可知，方程的根位于下限 r_l=0.5 与上限 r_u=1 之间，因此用两者的中点近似表示方程的根，即 r=0.75。

这可能相当准确了，但是我们可以采用更小的刻度，如图 6-6 所示。从这个图可知，方程的根位于下限 r_l=0.75 与上限 r_u=1 之间，因此，根的近似值 r=0.875。

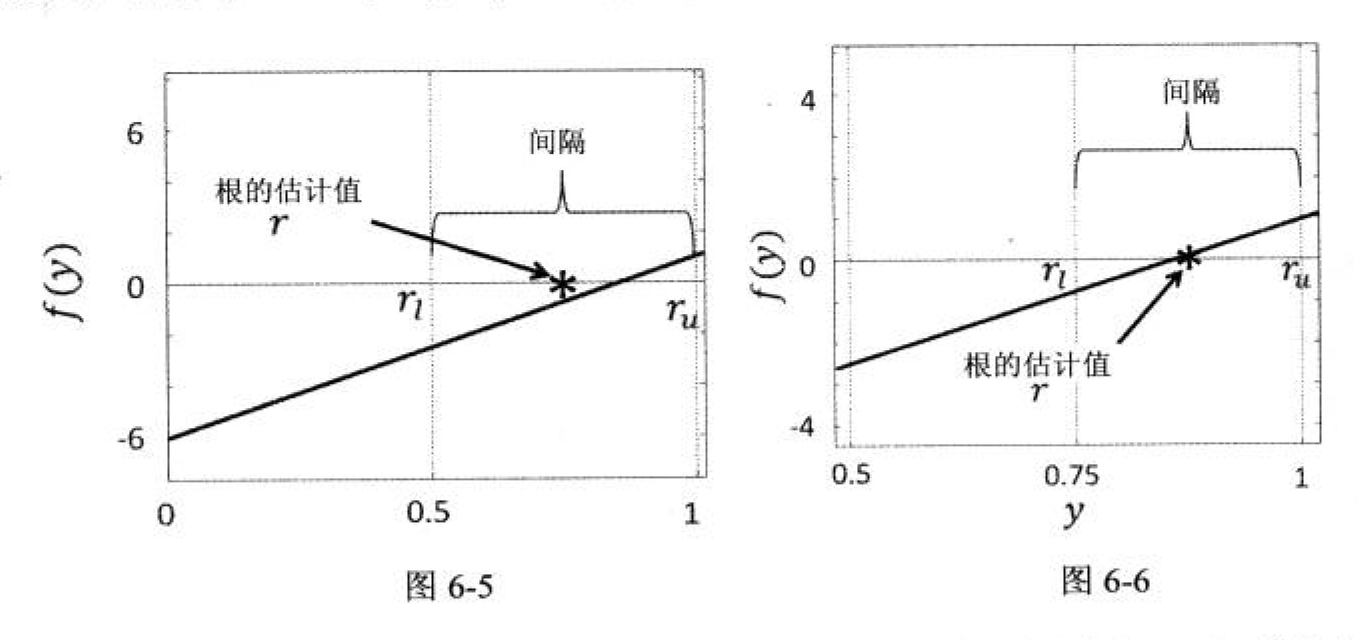


图 6-5

图 6-6

显然，我们可按这种方式，采用更精细的刻度，得到更加准确的解。但到现在为止，由于这是一个非常简单的线性方程，根本不需要用作图法，只需要简单的代数变换，我们就可以得到它的解析解，$y=\frac{6}{7}$，即 y=0.8571(或其他精度)。但是，仔细分析，我们发现作图法是我们理解方程求根的数值法的重要一步。事实上，本例中使用的方法就是算法解的二分法。它不断缩小变量两个端点之间的距离，这两个值的中点就是方程根的估算值。这个过程可以不断循环下去，直到我们认为它对手头要处理的工程问题足够准确为止。图 6-7 是二分法求根过程的流程图。在这个图里，我们利用 $f(r_l)$与 $f(r)$的乘积判断在这两个点之间是否存一个根。如果 $f(r_l)$和 $f(r)$有不同的符号，则表示方程的根肯定位于两个点之间的下半个区间里，否则方程的根位于此间隔的上半个区间里。

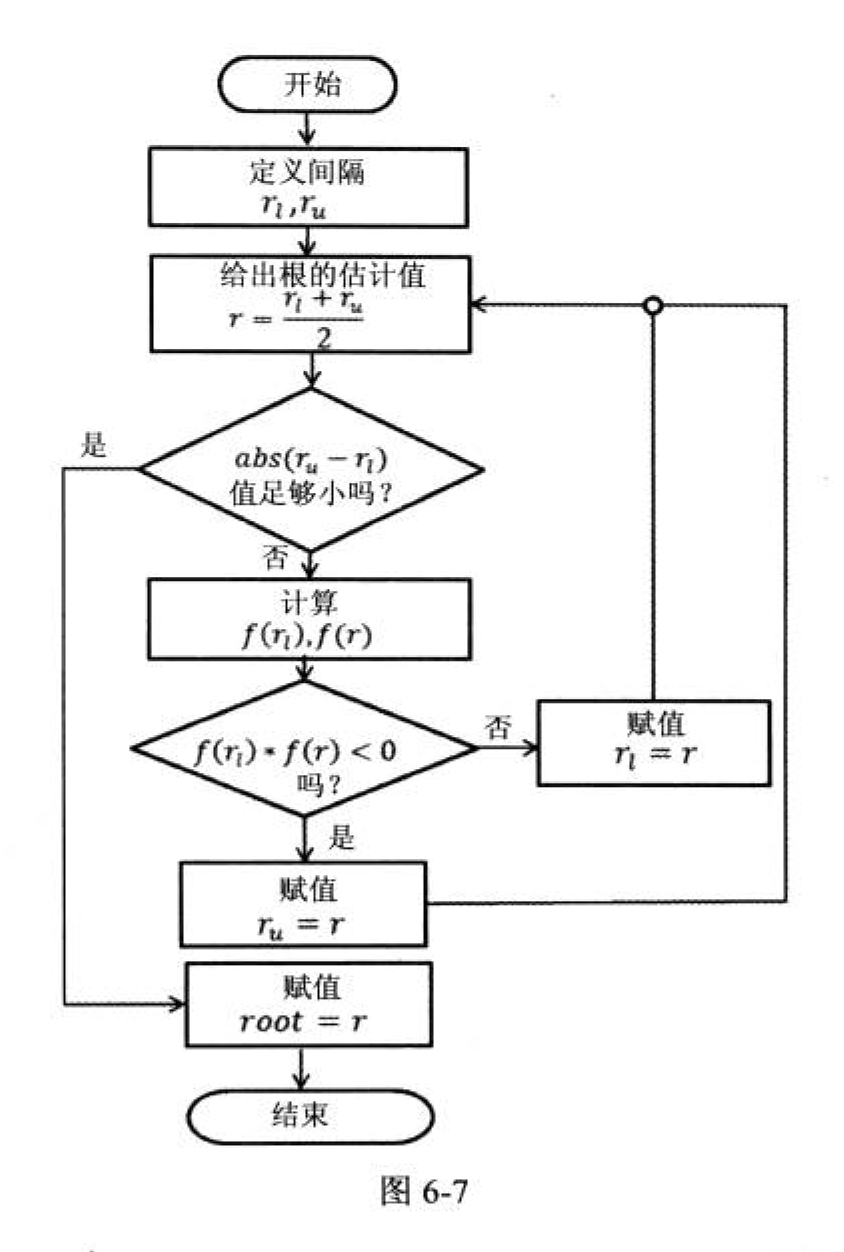


图 6-7

如果我们用 MATLAB 编写程序实现这个二分法算法，设两个初始点为 0 和 1，则根会收敛到一个非常准确的估算值。表 6-1 说明了计算过程。这个方程的求根需要 15 次迭代计算。

表 6-1　方程 6-18 二分法求根的迭代过程

y_l	y_u	y_r	$f(y_l)$	$f(y_r)$
0.0000	1.0000	0.5000	–6.0000	–2.5000
0.5000	1.0000	0.7500	–2.5000	–0.7500
0.7500	1.0000	0.8750	–0.7500	0.1250
0.7500	0.8750	0.8125	–0.7500	–0.3125
0.8125	0.8750	0.8438	–0.3125	–0.0938
0.8438	0.8750	0.8594	–0.0938	0.0156
0.8438	0.8594	0.8516	–0.0938	–0.0391
0.8516	0.8594	0.8555	–0.0391	–0.0117
0.8555	0.8594	0.8574	–0.0117	0.0020
0.8555	0.8574	0.8564	–0.0117	–0.0049
0.8564	0.8574	0.8569	–0.0049	–0.0015

(续表)

y_l	y_u	y_r	$f(y_l)$	$f(y_r)$
0.8569	0.8574	0.8572	–0.0015	0.0002
0.8569	0.8572	0.8571	–0.0015	–0.0006
0.8571	0.8572	0.8571	–0.0006	–0.0002
0.8571	0.8572	0.8571	–0.0002	0.0000

我们注意到，在这个二分法的流程图中使用了一个“迭代结束准则”。当间隔变得非常小时，程序终止，并输出最后得到的估算值。根据表 6-1 的结果，我们发现，当间隔的上下限之间小于 0.000 1 时，算法结束。当然，我们也可以继续迭代过程，直到求得的值与我们的“真实”解越来越接近。但是，程序必须在某个值时终止运行，这个值对于这个工程应用来说已足够准确了。

通过二分法的讨论结果，我们得出大多数求根算法的几个重要特征：

- 通常，这些求根算法需要在初始时先对根做某种形式的估算。例如在二分法中，这种估算就是一个其中包含根的取值范围(上限值和下限值)。在使用其他方法时，可能只需要在根的附近估算一个值。不管哪种情形，在试图求解之前需要知道根的大致位置。
- 根据预先提供的精度，确定这个根的误差值。在二分法中，算法规定最终的间隔值。在其他方法中，我们需要判断得到的估算值是否比较明显地改善了估算的精度。不管在哪种情况，重要的是必须记住：求根算法(或者说任何算法解方法)只能提供近似值，不能提供准确值。

有了这些知识之后，我们开始介绍方程求根的第二种算法。这种算法被称为牛顿迭代法(又称牛顿-拉夫逊法)。与二分法一样，这种求根方法需要估算方程的根，然后逐步改进，趋近于准确值。但不同于二分法的是，这种方法只需要一个估算值，利用函数在这个值斜率的投影，逐步趋向于准确值。

牛顿方法可以描述如下：假如我们要求函数 $y(x)$的根，先假设 x 的某个值，可以满足 $y(x)=0$，我们用 x_i 表示这个值。为了得到一个更接近根的估算值，我们要执行以下步骤：

- 计算函数 $y(x)$在这个位置的值，$y(x_i)$。
- 计算函数 $y(x)$在 x_i 位置的斜率，即 $y'(x_i)$。根据微积分知识，函数的斜率就是函数 $y(x)$的一阶导数。把 x_i 代入一阶导数公式，得到曲线 $y(x)$在 x_i 位置的切线的斜率，并求得该切线的方程，对照图 6-8，可以看出它们的关系。
- 该切线与 x 轴的交点比 x_i 接近方程的根。用 x_n 表示这个值，它可以用下面的公式求得：

$$x_n = x_i - \frac{y(x_i)}{y'(x_i)} \tag{6-19}$$

图 6-9 说明了上述式子的意义。

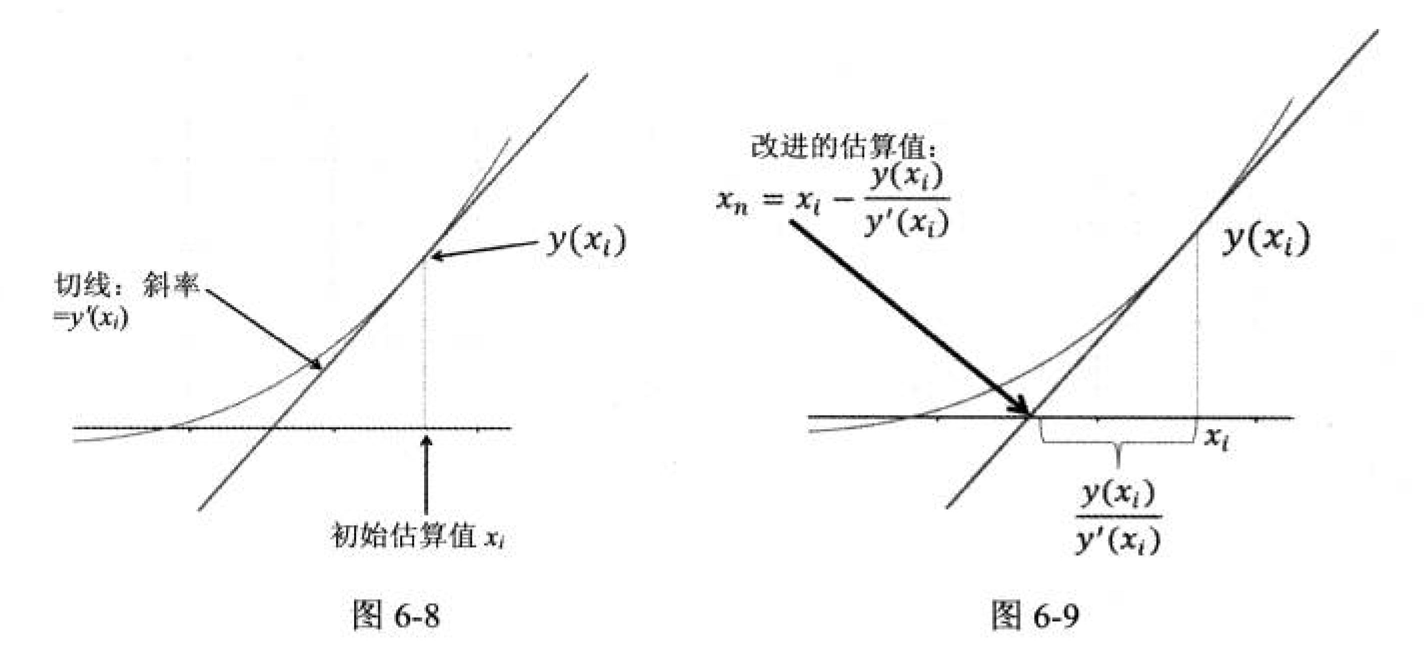


图 6-8　　　　　　　　图 6-9

反复执行上述步骤，直到我们认为求得的值足够接近于方程的根为止。

作为牛顿-拦夫逊算法的一个例子，我们考虑以下方程：

$$y = 3x^3 - 15x^2 - 20x + 50 \tag{6-20}$$

由微积分里的求导公式，该方程的在 x 位置的斜率按以下公式计算：

$$y' = 9x^2 - 30x - 20 \tag{6-21}$$

现在我们利用牛顿算法求方程(6-20)的根。

开始，我们任意取一个值 x=10，它是该方程的一个根，这完全是一个假设而已。当然，我们通过绘制方程(6-20)的曲线，可以设置一个比较准确的猜测值。但是在这个例子里，任意一个猜测值都可以。现在我们应用牛顿迭代法：

- 求方程(6-20)在 x_i=10 位置的值，在本例，它的值是：

$$\begin{aligned} y(10) &= 3(10)^3-15(10)^2-20(10)+50 \\ &= 1350 \end{aligned} \tag{6-22}$$

- 把 x_i=10 代入方程(6-21)，计算曲线在该点 x_i=10 位置的斜率。在本例中，这个斜率的值是：

$$\begin{aligned} y' &= 9(10)^2-30(10)-20 \\ &= 580 \end{aligned} \tag{6-23}$$

- 根据公式(6-19)，得到一个更接近方程根的估算值，在本例中，这个值是：

$$\begin{aligned} x_n &= 10 - \frac{1350}{580} \\ &= 7.6724 \end{aligned} \tag{6-24}$$

图 6-10 说明了上述步骤。

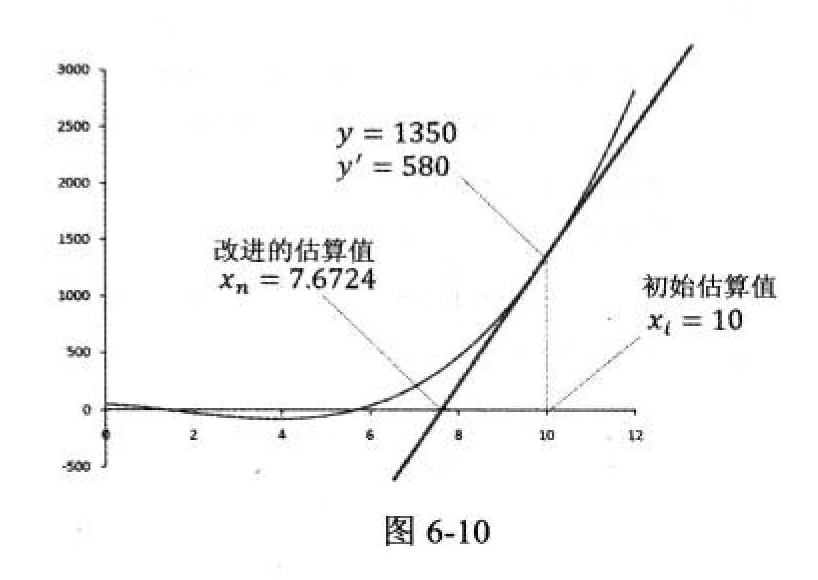


图 6-10

把上述的估算值代入方程(6-20)，得到 y 的值为 368.494，如果我们认为这个值很接近于 0，则算法结束，如果还不够接近于 0，则把 7.6724 赋给 x_i，重复上述过程。表 6-2 列出了牛顿迭代法的 7 次迭代过程。注意，从第 5 次迭代后，根的最后 4 位小数都相同。

表 6-2　方程(6-20)牛顿算法的 7 次迭代过程

Xi	y	y'	xn
10.000 0	1 350.00	580.000 0	7.672 4
7.672 4	368.494 1	279.621 0	6.354 6
6.354 6	87.004 9	152.788 7	5.785 1
5.785 1	13.127 3	107.655 9	5.663 2
5.663 2	0.545 7	98.750 2	5.657 7
5.657 7	0.001 1	98.352 9	5.657 7
5.657 7	0.000 0	98.352 1	5.657 7

图 6-11 是这个方法的流程图。

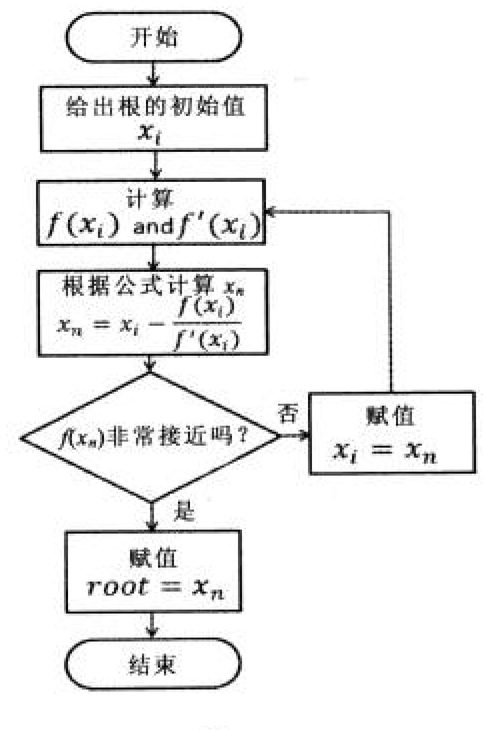


图 6-11

比较二分法与牛顿迭代法，我们发现其中一个主要的差别是：

- 二分法需要两个初始估算值，而且方程的根必须位于这两个估算值之间。因此我们称二分法为插入法(bracketing method)。
- 牛顿迭代法只要求一个初始估算值，而且对它的位置也没有特别的要(唯一的要求是它的斜率不要为0)。因此我们称它为开型法(open method)。

插入法和开型法还有其他一些重要的差别：

- 对于插入法，只要在两个初始值之间存在一个根，这种算法就保证能得到方程的根，不管最终的精度要求如何。正因为如此，我们认为插入法可保证收敛。开型法不能保证求得一个根；这种方法有时会发散，即算法不能在某个根的附近点终止。因此，程序员在使用开放算法时，必须注意它的发散性，当发现发散时，就停止执行。
- 虽然开型法不能保证收敛，但是在大多数情况下，开型法的收敛速度比插入法的收敛速度快许多。使用开型法，用较少的迭代次数就可以达到满意的精度。
- 当方程有多个根时，用插入法总会找到存在于下限与上限之间的一个根。开型法会收敛到某一个根，但是有时我们无法预测到开型法收敛到哪一个根。通常，开型法收敛到与初始值最接近的一个根，但是情况并非总是如此。这要取决于曲线的形状。
- 尽管这两种方法都要反复计算函数的值，但是牛顿法需要求函数的斜率，即需要计算函数的导数。这需要使用微积分公式或者使用数值法求导数。

虽然插入法有保证收敛的优点，但是，大多数商业求根程序使用开型法。

为了说明两者之间的差别，我们用二分法求解方程(6-20)，表6-3里说明了求解结果，算法中使用的初始猜测值为 x_l=5.000 0，x_u=6.000 0。与表6-2相比较，我们发现，虽然二分法也收敛到牛顿迭代法的同一个结果，但是它需要15次迭代过程。

在本书的剩下部分，我们将介绍如何用MATLAB和Excel求解代数非线性方程。在此过程中，必须记住有关求根算法的几个结论。

表6-3 用二分法求方程(6-20)的15次迭代过程

x_l	x_u	x_r	$y(x_l)$	$y(x_r)$
5.000 0	6.000 0	5.500 0	–50.000 0	–14.625 0
5.500 0	6.000 0	5.750 0	–14.625 0	9.390 6
5.500 0	5.750 0	5.625 0	–14.625 0	–3.173 8
5.625 0	5.750 0	5.687 5	–3.173 8	2.967 0
5.625 0	5.687 5	5.656 3	–3.173 8	–0.138 5
5.656 3	5.687 5	5.671 9	–0.138 5	1.405 5
5.656 3	5.671 9	5.664 1	–0.138 5	0.631 3
5.656 3	5.664 1	5.660 2	–0.138 5	0.245 9
5.656 3	5.660 2	5.658 2	–0.138 5	0.053 6
5.656 3	5.658 2	5.657 2	–0.138 5	–0.042 5

(续表)

x_l	x_u	x_r	$y(x_l)$	$y(x_r)$
5.657 2	5.658 2	5.657 7	−0.042 5	0.005 5
5.657 2	5.657 7	5.657 5	−0.042 5	−0.018 5
5.657 5	5.657 7	5.657 6	−0.018 5	−0.006 5
5.657 6	5.657 7	5.657 7	−0.006 5	−0.000 5
5.657 7	5.657 7	5.657 7	−0.000 5	0.002 5

6.3　教程：用 MATLAB 求解普通非线性方程的根

本节将介绍如何用 MATLAB 求解普通的非线性方程。实际上，本节介绍的方法可以应用于任何代数方程。但是，由于求解多项式方程有更简单和更强大的工具，因此，本节介绍的方法最适合于普通非线性方程。

求解普通非线性方程的步骤归纳如下：

(1) 把方程写成标准的 $f(x)$。

(2) 设计一个 MATLAB 函数，输入 x，求 $f(x)$。

(3) 用 fplot 命令，求得方程一个根的估算值。

(4) 用 fzero 命令(以及前面求得的估算值)计算方程的根。

(5) 测试方程的根，保证满足 $f(x)=0$。

在下面的例子里，我们将详细介绍这个求解过程的每个步骤。

例 6.1

求满足以下表达式的 x 值：

$$x \tan x = 0.4 \tag{6-25}$$

解：

步骤 1：把方程写成标准形式。还记得什么是方程的标准形式吗？所谓的标准形式是把方程转换为 $f(x)=0$ 的形式。因此有：

$$f(x) = x \tan x - 0.4 \tag{6-26}$$

步骤 2：编写一个 MATLAB 程序，计算 $f(x)$的值。在 MATLAB 的编辑器里，建立以下函数，并把它保存为 ex6_1。

```
1 function f = ex6_1(x)
2 f = x*tan(x) - 0.4;
```

对于任意的输入值 x，这个函数可以求 $f(x)$。在命令提示符后，输入以下命令，验证此函数：

```
>> ex6_1(4)
ans =
      4.2313
```

步骤3：用 fplot 命令求得该方程根的一个估算值。本例使用的求根算法需要在所求根的附近确定一个初始值。由于我们已经把这个方程定义为一个 MATLAB 函数，因此可以用智能的 fplot 命令画出函数的图形(画图的方法在第5章中曾介绍过)。从这个图里，我们可以估算根的值。但是在画图之前，需要大致确定 x 的一个范围，在此范围内方程肯定会有一个根。我们先尝试取值范围为 0~1.5。在命令提示符后输入以下命令：

```
>> fplot ('ex6_1',[0,1.5]
```

图 6-12 是该方程的曲线。

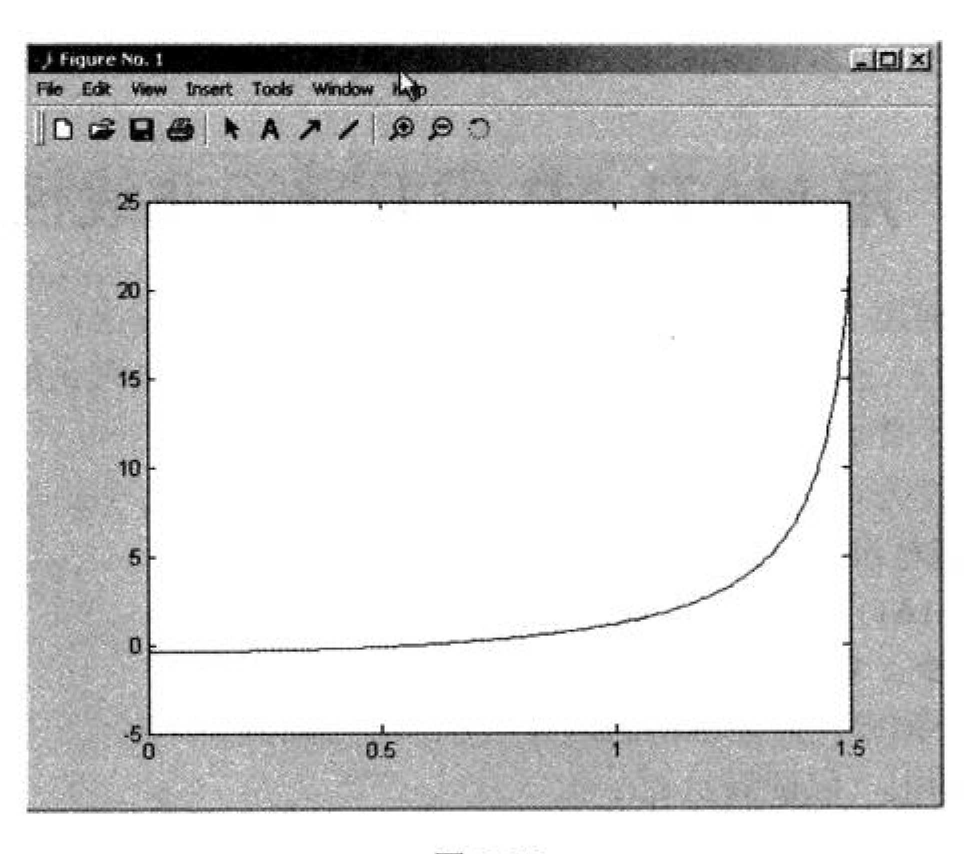


图 6-12

很幸运，在这个范围内存在一个根(如果根不存在，可以加大这个取值范围，或者改变图形间隔)。仔细观察这个曲线，我们估算，方程的根在 0.5 附近。

步骤4：用 fzero( )和估算值，求方程的根。fzero( )是 MATLAB 内置的求根函数。它的用法是：

fzero('函数名',估算值)

fzero( )返回方程(由 function_name 定义)在初始估算值附近的一个根。注意，fzero( )只需要一个估算值。由此可知，它使用开放法求方程的根。输入以下命令，可以求得 ex6_1 函数描述的方程在 0.5 初始估算值附近的根，并把求得的根保存在变量里：

```
>> r = fzero('ex6_1',.5)
r =
      0.5932
```

返回的 0.593 2 就是该方程的根。我们应该知道，这个方程可能会有多个根。这里我们只是求得离初始值最近的一个根。

步骤5：把求得的根代入 $f(x)=0$ 进行验证。读者要养成一个良好的习惯——验证求得的结果。在本例的这个问题里，我们需要把最后得到的根代入函数，看看它的结果是否为

0。由于方程的根已保存在变量 r 里，而且方程已用一个函数程序表示，因此，在命令窗口输入以下命令可以验证方程的根：

```
>> ex6_1(r)
ans =
  -5.5511e-017
```

我们注意到，函数的值确实近似等于 0。

这 5 步——即以 fzero( )作为求根命令的方法——可以用来求解任意代数方程的根。但是由于 MATLAB 提供了更加简单、功能更强的多项式方程的求根工具，因此，我们通常用这 5 步法用来求解普通非线性方程。

6.4 教程：用 MATLAB 求解多项式方程的根

本节介绍的方法最适合于多项式方程的求解。该方法使用 MATLAB 的 roots 命令，它可以求得一个多项式方程的全部解(包括实根和虚根)，而且不需要设置初始值。

该求解方法的步骤如下：

(1) 把方程写成标准形式 $f(x)=0$。

(2) 在 MATLAB 里定义一个数组，表示多项式的系数常量。

(3) 用 MATLAB 的 roots 命令求多项式方程的根。

(4) 用 MATLAB 的 polyval 命令验证结果。

在下面的例子里，我们将详细介绍该方法的求解步骤。

例 6.2

求满足以下表达式的一个 x 值：

$$2x^2+10x=\frac{144}{x} \tag{6-27}$$

解：

步骤 1：把方程写成标准形式。还记得什么是方程的标准形式吗？所谓的标准形式是把方程转换为 $f(x)=0$ 的形式。此外，由于多项式的自变量 x 的次数只能正整数，因此，方程必须排列成如下形式：

$$f(x)=2x^3+10x^2-144$$

注意方程中自变量 x 的次数必须从左到右递减。本例中，第一项是 x^3，最后一项是 x^0(常量)。这样排列便于进行步骤 2 的处理。

步骤 2：定义一个 MATLAB 数组，表示多项式方程的系数常量。为了简化多项式方程的计算，MATLAB 内置了功能强大的函数。为了使用这些函数，我们必须定义一个行数组，它保存多项式方程的各个系数。例如，对于如下形式的多项式：

$$f(x)=A_1x^n+A_2x^{n-1}+A_3x^{n-2}+\cdots+A_{n-1}x+A_n \tag{6-28}$$

则表示这个多项式的数组必须是：

$$p = [A_1\ A_2\ \cdots A_{n-1}\ A_n\ ] \tag{6-29}$$

在 MATLAB 的命令窗口输入以下命令，定义多项式方程的系数数组。

```
.>> p = [2 10 0 -144];
```

注意，与 x^1 项的系数对应的元素必须输入 0。

步骤 3：用 MATLAB 的 roots 命令求多项式方程的根。roots 命令是 MATLAB 的一个功能强大的计算工具，它可以求多项方程的全部解。下面是它的用法：

roots(coefficient_array)

这里的 coefficient_array 就是多项式的系数数组。在 MATLAB 命令提示符后输入以下命令，可以求这个多项式方程的根：

```
>> r = roots(p)

r =

   -4.0000 + 2.8284i
   -4.0000 - 2.8284i
   3.0000
```

输出该多项式方程的全部 3 个根。把它们保存在数组 r 里。注意，其中两个是虚根。

步骤 4：用 MATLAB 的 polyval 命令验证结果。读者要养成一个良好的习惯——验证结果。在方程求根过程中，我们必须计算方程在每个根的值，这个值必须是 0。既然，我们用一个数组表示多项式方程，因此可以用 MATLAB 的 polyval 命令计算多项式在自变量 x 位置的值。下面是 polyval( )函数的用法：

polyval(coefficient_array, value)

其中，coefficient_array 是多项式方程的系数数组，*value* 是自变量 x 的值，polyval 求得多项式在该位置的值。例如，根据步骤得到多项式系数数组，输入以下命令：

z=polyval(p,2.5)

求得方程(6-28)在 x=2.5 的值，结果是 9.75，并把结果保存在变量 x 里。

由于我们已经求得多项式方程的根，并且保存在变量 r 里，因此很容易求得多项式在这些根位置的值，并验证这个值是否为 0。在命令提示符后输入以下命令：

```
>> polyval(p,r)
ans =
    1.0e-012 *
   -0.1990 - 0.0426i
   -0.1990 + 0.0426i
         0
```

我们发现，在这些根位置，多项式的值都非常接近于 0。

需要说明的是，多项式方程当然也可以用第 6.3 节里介绍的方法求解。但是，每个根需要单独求解，而且求虚根非常不容易。本节介绍的方法是解多项式方程的首选方法。

6.5　教程：用 Excel 求解普通非线性方程的根

Excel 也可以用来求普通非线性方程的根。它有两个工具可以求方程的根。一个是单变量求解(Goal Seek)，另一个是 Solver。前者比较简单，本节将介绍这个工具。后者比较复杂，将在第 8 章和第 10 章介绍。

Goal Seek 使用的算法与 MATLAB 里使用的 fzero 命令相似。需要先定义要求解的方程，然后定义一个初始估算值，单变量求解只能在估算值邻域里求得一个根。我们再通过例 6.1 来说明单变量求解的用法。

例 6.3

求满足以下表达式的 x 值

$$f(x) = x\tan x - 0.4 = 0 \tag{6-30}$$

解：

先新建一个 Excel 表格，根据图 6-13 所示输入 A1 和 A2 单元的内容。

在单元 B1 里，输入根的初始估测值。如果我们对这个函数毫无所知，则可在这一步画出这个函数的曲线，利用曲线确定初始解。但是由于我们前面已经介绍过这个函数，因此使用前面的 MATLAB 例子里的初值(x=0.5)。在 B1 单元里输入 0.5，在 B2 单元里输入方程(6-30)，引用 B1 单元作为它的值，如图 6-14 所示。按 Enter 键接受此公式。

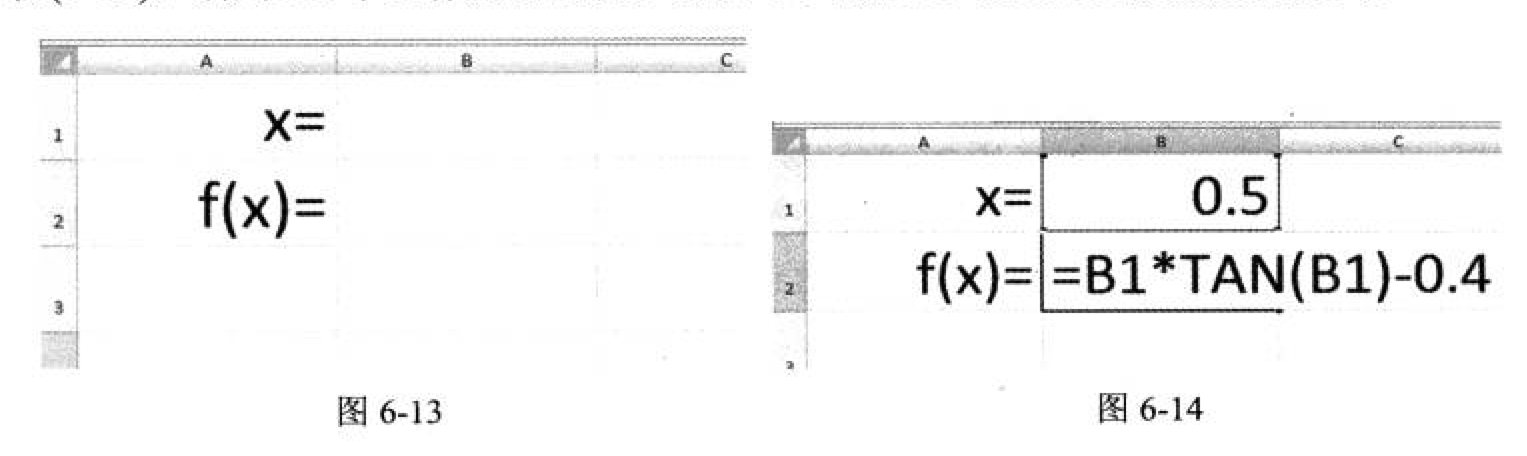


图 6-13　　　　图 6-14

单击单元 B2，选择 Ribbon 的数据选项卡，在【假设分析】下拉菜单中选择【单变量求解】命令，如图 6-15 所示。

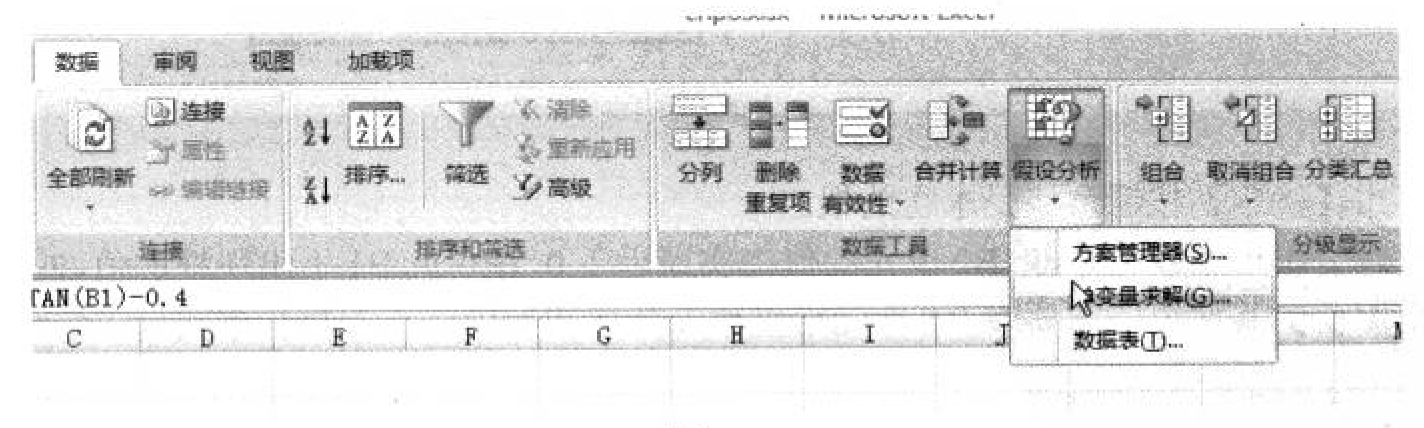


图 6-15

弹出【单变量求解】对话框，如图 6-16 所示。

由于在此之前我们已选择单元 B2，因此在这个对话框的【目标单元格】文本框里已有

B2 单元。它就是求解方程所在的单元。单击【目标值】文本框，输入 0，如图 6-17 所示。

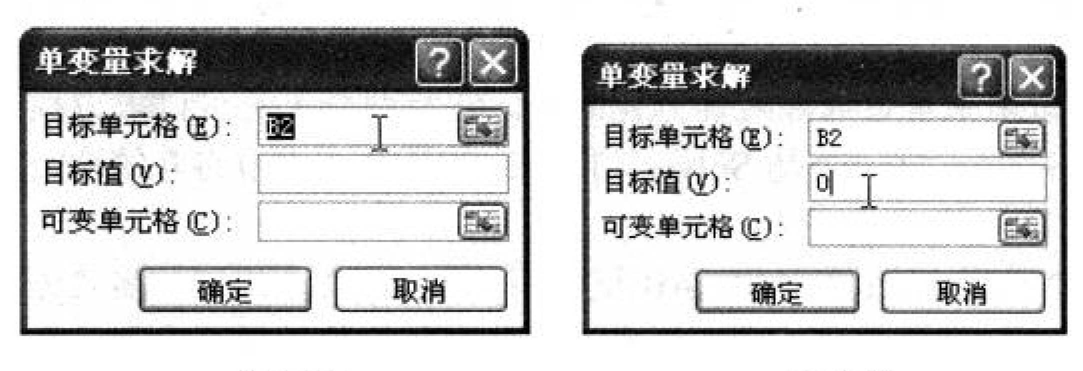


图 6-16　　　　图 6-17

它就是 B2 单元里的方程的目标值，我们把这个目标值设置为 0(即相当于 $f(x)=0$)。单击【可变单元格】的文本框，选择 B1 单元格，如图 6-18 所示。

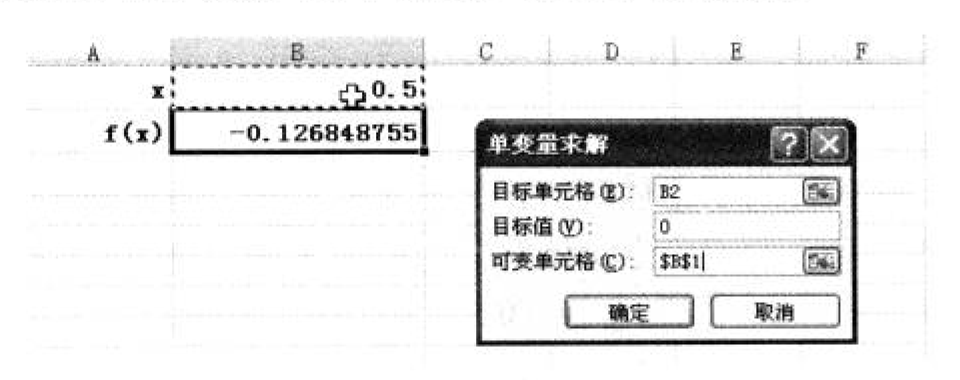


图 6-18

上述设置表示，我们把 B1 设置为方程的自变量。

Excel 中的单变量求解程序从 B1 单元格里的值出发，开始搜索单元格 B2 方程的一个根，直到方程值到达目标值为止。单击【单变量求解】对话框里的【确定】按钮，就可开始计算。单变量求解程序开始执行，直到达到目标值时才终止，如图 6-19 所示。

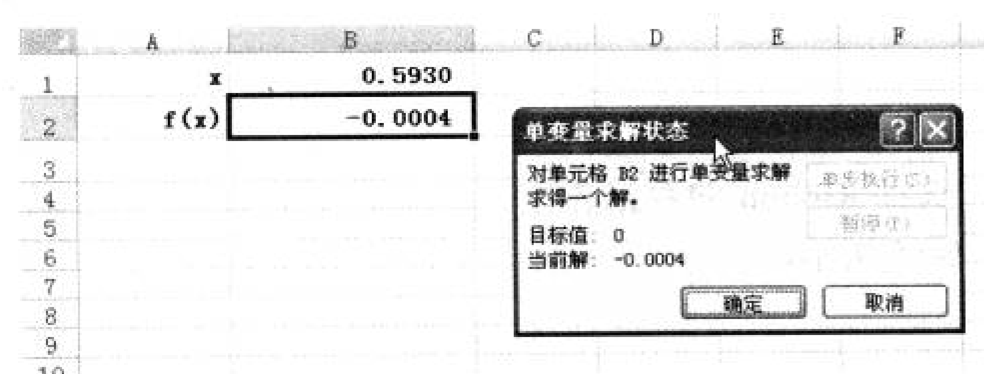


图 6-19

单击【确定】按钮，接受求得的根。

通过分析电子表格可知，B2 单元的值确实逼近于 0，在单元 B1 里出现根的值，即 0.593。要求出方程的其他根，需要在 B1 单元格里输入其他初始值，再次执行单变量求解程序。

6.6 习题

1. 为以下每个方程和自变量的取值范围，写一个 MATLAB 函数，用 MATLAB 的 fzero

命令求方程的根。

2. 用 Excel 的单变量求解方法重做习题 1。

3. 用下面的方法求解第 6.1 节中的横梁问题。

(a) 用 MATLAB 的多项式求解方法。

(b) 用 MATLAB 的普通非线性方程求解方法。

(c) 用 Excel 的普通非线性方程求解方法。

讨论每种方法的结果及适用性。

4. 根据下面的方程：

$$3x^4 - 6x = 6$$

(a) 把这个方程转换成为标准形式，用 MATLAB 的绘图命令绘制它的曲线。

(b) 编写一个简单的程序，用二分法求出该方程的任意一个根，要求精度在 0.01。

(c) 编写一个简单的 MATLAB 脚本程序，用 roots 命令求解这个多项式方程的根。

(d) 用 Excel 的单变量求解工具验证上述结果。

5. 编写一个 MATLAB 脚本程序(要求使用函数)，求以下方程的普通非线性代数方程的第一个正根和第一个负根(绝对值最小的两个根)，要求使用 fzero 命令。

$$7x = 4.5 + x^3 \cos x$$

6. 用 Excel 的单变量求解工具重做习题 5。

7. 根据第 6.2 节介绍的方法，在一个电子表格里使用二分法求以下方程的一个根：

$$y = 5x^2 - 12x + 3$$

初始取值范围为 $0 \leqslant x \leqslant 1$，继续分割，直到根的误差在 0.001 为止。把初始取值范围改为 $1 \leqslant x \leqslant 3$，重复上述过程，求另一个根。利用二次方程求根公式，手工验证自己的结果。

8. 在 Excel 里，用牛顿迭代法重做习题 7。提示，上述函数的导数为：

$$y' = 10x - 2$$

9. 一个被称为四杆机构的机械装置如图 6-20 所示。在机械系统中使用这样的装置会根据输入动作生成相应的输出动作。这个四杆装置只有一个自由度。如果其中任意一个链杆位置已知，可以导出其他链杆的位置。假设一个电机驱动链杆 2，链杆 2 的角度 θ_2 已知，则由以下方程可以求得链杆 4 的角度 θ_4：

$$L_1 = L_4 \cos\theta_4 - L_2 \cos\theta_2 + \sqrt{L_3^2 - (L_4 \sin\theta_4 - L_2 \sin\theta_2)^2}$$

式中，L_i 是链杆 i 的长度(注意，链址 1 是“隐含链杆”，它表示两个旋转支点之间的距离)，假设 L_1=6in，L_2＝4in，L_3=7in 和 L_4=5.2in。根据以上条件，计算：

(a) 用 MATLAB 计算，输入角度 θ_2 为多大时，才会使输出角度 θ_4＝80°。

(b) 用 Excel 验证(a)中的结果。

(c) 按比例画出这个机械装置图，用 CAD 程序或手工验证(a)和(b)的结果。

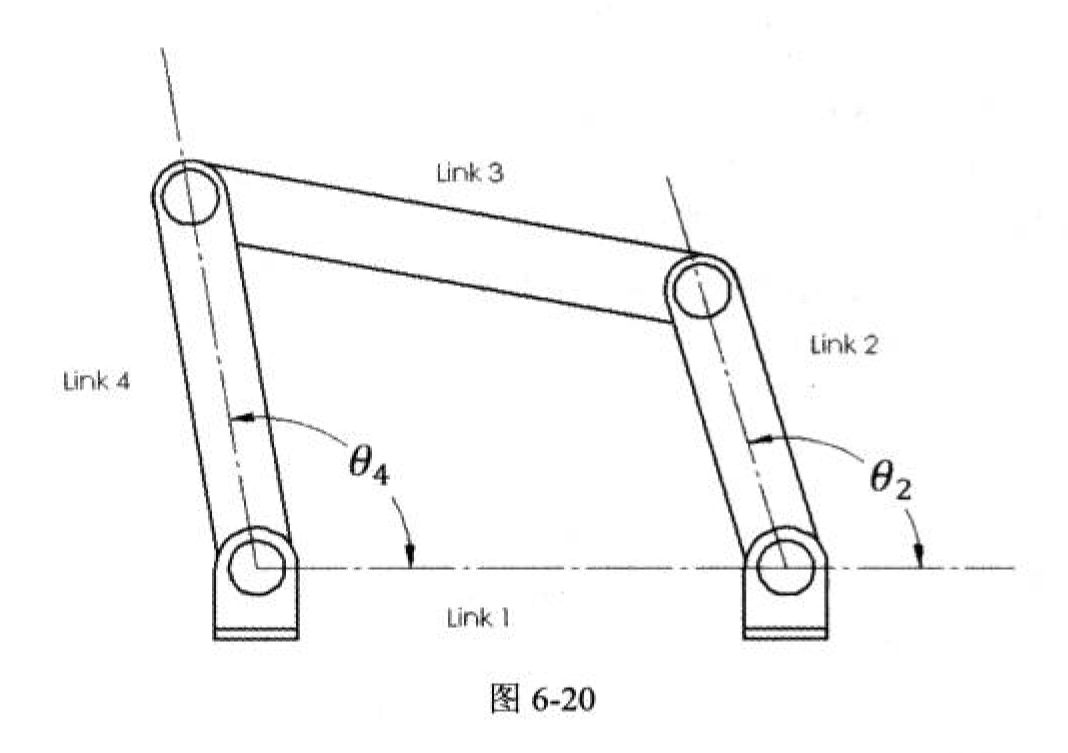


图 6-20

10. 假设年利率为 i(表示为小数值，不是一个百分比)，存入的金额为 P，则 n 年后，这笔投资的总金额可以由以下公式求得：

$$A = P(1+i)^n$$

(a) 设计一个 MATLAB 程序，要求：

- 用户输入一个最初的投资金额 P，投资的目标值 A 和年利率 i。
- 求出多少年后，这笔初始投资额会增长到目标值(用方程求根方法)，输出结果。

用初始投资额 1000 0，目标值为 1000 00，年利率为 0.10，测试此程序。

(b) 用 Excel 的单变量求解工具验证 a 的结果。

(c) 针对上述的测试值，绘制一个曲线表示 A 与 n 的关系，根据此曲线验证(a)和(b)的结果。

11. 重做习题 6.11 的利息计算。

(a) 编写一个 MATLAB 程序，要求：

- 用户输入一个最初的投资金额 P，投资的目标值 A 和投资年限 n。
- 年利率为多大时，这笔初始投资额经过 n 年后，增长到目标值(用方程求根方法)，输出结果。测试此程序。

(b) 用 Excel 的单变量求解工具验证 a 的结果。

(c) 针对上述的测试值，绘制一个曲线表示 A 与 n 的关系，根据此曲线验证 a 和 b 的结果。

12. 一个机械系统按照以下模型振动：

$$y(t) = 7e^{-0.09t}(\sin(1.15t) + \cos(1.15t))$$

式中，$y(t)$表示振动的位置，单位为 in，t 是时间，单位为 s。振动的周期就是完成一个振动循环所需要的时间。按照下面的要求，用 MATLAB 或 Excel 计算该振动系统的周期：

(a) 画出振动位置随时间变化的函数。

(b) 用方程求根方法求该振该动系统第一次经过零位置的时间。

(c) 用方根求根方法求该振动系统的振幅第三次为零的时间。

(d) 根据两次振动的时间差估算振动周期。

振动周期与公式中的一个参数有关，读者根据自己的观察，确定哪个参数与周期有关，并验证自己的答案。

13. 针对第 5 章习题 16 的升降机问题，绳子通一个圆柱滑轮提升一个重为 500N 的物体，绳子与滑轮的摩擦系数为 μ=0.3。一位工程师打算在这个装置里使用绳子升降物体。假如绳子的最大负荷为 750N。根据以下要求，求出最大的接触角度 θ 为多大？

(a) 把绳子的拉力公式变成标准形式。

(b) 用 MATLAB 画出这个方程的曲线。

(c) 用 MATLAB 的 fzero 命令求出这个最大接触角度 θ。

(d) 用 Excel 的单变量求解工具验证结果。

14. 针对第 5 章习题 16 的升降机问题，绳子通一个圆柱滑轮提升一个重为 500N 的物体，绳子与滑轮的摩擦系数 μ 未知。一位工程师根据实验结果发现，当接触角度 θ 为 85º 时，要把一个重为 500N 提起来，绳子的拉力为 950N。根据下面的要求，计算摩擦系数：

(a) 把绳子的拉力公式变成标准形式。

(b) 用 MATLAB 画出这个方程的曲线。

(c) 用 MATLAB 的 fzero 命令求出这个最大接触角度 θ。

(d) 用 Excel 的单变量求解工具验证结果。

15. 在第 2 章我们介绍了轴承的焠火过程。在滚球轴承的制造过程中，各部件需要经过一道硬化工序。具体过程是先加热再迅速冷却，或者把它浸入到油槽或水槽里，这个过程即为淬火。滚球轴承的温度是时间的函数。$T(t)$可以按方程(2-3)估算：

$$T(t) = (T_i - T_\infty)e^{-\frac{t}{\tau}} + T_\infty$$

式中 t 是轴承浸入槽中的时间，单位为 s。T_i 是轴承的初始温度，T_∞ 是油的温度。τ 是时间常数，单位是 s，它与轴承的材料、轴承的几何形状及油的特性有关。假设轴承的初始温度为 800ºC，油的温度为 40ºC，时间常数 τ ＝50s，选择任意一种方法，计算轴承的温度降低到 600ºC、300ºC 和 50ºC 时所需要的时间。

第 7 章

矩阵运算

引言

矩阵是数学量的一维或二维数组的表示。矩阵的定义、表示和数学运算是解决工程问题的一项重要技术。在工程中，矩阵理论经常应用于控制系统、结构分析、热力学和流体力学中。有关利用矩阵解决这些问题的一些例子将在第 8 章中介绍。

在本章中，我们将学习以下内容：

- 如何在 Matlab 和 Excel 里表示矩阵。
- 如何在 Matlab 和 Excel 里进行矩阵加法运算。
- 如何在 Matlab 和 Excel 里进行矩阵乘法运算。
- 如何在 Matlab 和 Excel 里进行矩阵与标题的乘法运算。
- 如何在 Matlab 和 Excel 里求矩阵转置。
- 如何在 Matlab 和 Excel 里求矩阵的秩。
- 如何在 Matlab 和 Excel 里求矩阵的逆。

7.1 矩阵的性质

本节将向读者概括性地介绍矩阵的基本性质，这样，读者就掌握了如何在 MATLAB 和 Excel 里进行矩阵的基本运算。这个问题的详细而严密的讨论将留在数学教材里进行。

矩阵可以看作是一组数的有规律排列，它表现为表格的形式，如图 7-1 所示。矩阵常用一个名字来表示，在本例中，我们用 A 表示这个矩阵。注意，通常矩阵名使用黑体字体。

$$\mathbf{A} = \begin{bmatrix} 5 & 3 \\ 2 & 8 \end{bmatrix}$$

图 7-1

矩阵的每一行和每一列都用一个索引值表示。通常我们用行 1 表示矩阵 **A** 的第一行，用行 2 表示矩阵的第 2 行，等等。同样道理，列 1 和列 2 分别被称为矩阵的第 1 列第 2 列。用元素表示矩阵里的每一个数。元素通常用 a_{ij} 表示，这里的 i 表示它的行号，j 表示列号。采用这种表示方法，我们可以把矩阵表示成如图 7-2 的形式。

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix}$$

图 7-2

读者可能已经注意到，矩阵 **A** 有两行两列，而矩阵 **B** 有三行两列。矩阵的行数和列数表示矩阵的阶。前述的矩阵 **A** 是 2×2 阶矩阵，**B** 是 3×2 阶矩阵。矩阵 **A** 也被称为方阵，因为它的行数与列数相等。掌握矩阵阶的概念非常重要，因为矩阵的许多运算的结果都与矩阵的阶有关。

7.1.1　矩阵相加运算

阶相等的多个或两个矩阵可以相加，相加公式用 **C**＝**A**＋**B** 表示，其中 **C** 被称为矩阵 **A** 与矩阵 **B** 的和。矩阵的加法运算可以用 $c_{ij}=a_{ij}+b_{ij}$ 公式(见公式(7-1))表示，如图 7-3 所示。矩阵的加法运算是可交换的，即 **A**＋**B**＝**B**＋**A**。

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} + \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{7-1}$$

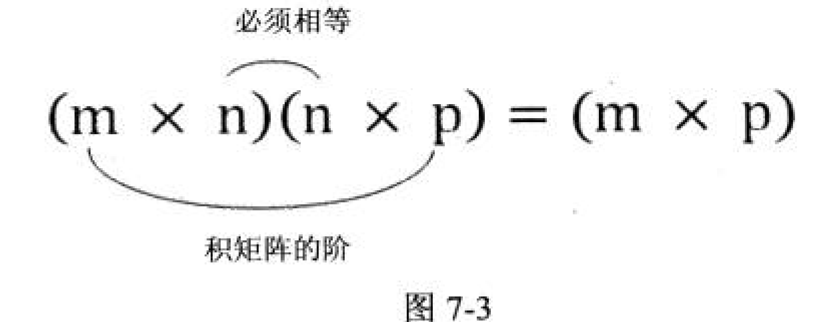


图 7-3

7.1.2　矩阵相乘运算

两个矩阵能否相乘运算取决于两个矩阵的阶和它们的运算顺序。矩阵的相乘运算用 **C=A×B**，或者用 **C=AB** 表示。只有当第一个矩阵的列数等于第二个矩阵的行数时，两个矩阵才可以进行相乘运算。积 **C** 的阶数由第一个矩阵的行数和第二个矩阵的列数决定。图 7-3 用(m×n) (n×p)= (m×p)说明两个矩阵相乘运算的阶数的要求以及积的阶数。

矩阵运算的过程是这样的：第一个矩阵第一行的每个元素与第二个矩阵的第一列相应的元素相乘，然后把相乘结果相加，得到 C_{11}，其他元素按同样方法进行。第一个矩阵的第一行与第二个矩阵的第二列，得到 C_{12}，依次类推，直到第一个矩阵的每一行都与第二个矩阵的每一列相乘为止。为了更好地理解矩阵的运算过程，可以对照式子(7.2)~(7.6)。这

些公式详细说明了一个 2×3 矩阵如何与一个 3×2 矩阵相乘。

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{212} & b_{23} \end{bmatrix}=\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \tag{7-2}$$

式中：

$$c_{11}=a_{11}b_{11}+a_{12}b_{21}+a_{13}b_{31} \tag{7-3}$$

(第一行乘上第一列，得到 c_{11})

$$c_{12}=a_{11}b_{12}+a_{12}b_{22}+a_{13}b_{32} \tag{7-4}$$

(第一行乘上第二列，得到 c_{12})

$$c_{21}=a_{11}b_{12}+a_{22}b_{21}+a_{23}b_{31} \tag{7-5}$$

(第二行乘上第一列，得到 c_{21})

$$c_{22}=a_{21}b_{12}+a_{22}b_{22}+a_{23}b_{32} \tag{7-6}$$

(第二行乘上第二列，得到 c_{22})

矩阵的乘法运算是不可交换的，即 $\mathbf{AB}\neq\mathbf{BA}$。

7.1.3 矩阵与标量的乘法运算

一个矩阵乘上一个标量，就是把矩阵的每个元素进行放大或缩小，得到一个新矩阵。矩阵与标量的乘法运算用 $\mathbf{C}=k\mathbf{A}$ 表示。其中，k 是任意一个标量。矩阵与标量的乘法运算时矩阵的每个元素都乘上这个标量值，公式 7-7 说明了其运算规律：

$$k\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}=\begin{bmatrix} ka_{11} & ka_{12} \\ ka_{21} & ka_{22} \end{bmatrix} \tag{7-7}$$

7.1.4 单位矩阵

单位矩阵用 **I** 表示，它是一个方阵，它的一个重要性质是，任何矩阵 **A** 乘上单位矩阵 **I**(假设它们的阶数符合乘法要求)，结果还是矩阵 **A**。从这个意义上讲，一个矩阵乘上 **I** 相当于乘上标量 1。因此矩阵与单位矩阵的相乘可以表示为

$$\mathbf{IA}=\mathbf{A} \tag{7-8}$$

当矩阵与单位矩阵相乘时，它有一个独特的性质是满足交换律，即 $\mathbf{IA}=\mathbf{AI}=\mathbf{A}$。

单位矩阵同时也是一个对角矩阵，对角矩阵是指除从左上角到右下角的对角线上的元素外，其他元素都为 0。因此，单位矩阵是一个特殊的对角矩阵，它的对角线的元素都为 1。图 7-4 是一个 3×3 的单位矩阵。

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

图 7-4

7.1.5 矩阵的转置

有时需要转置一个矩阵。一个矩阵的转置是指重新排列矩阵的行列元素，使得第 1 行变为第 1 列，第 2 行变为第 2 列，等等。任何矩阵都可以进行转置运算，转置运算会改变矩阵的阶数，**A** 的转置矩阵常用 $\mathbf{A}^{\mathrm{T}}$ 表示。图 7-5 说明了矩阵的转置算法。注意，原来是 2×3 的矩阵现在变成了 3×2 的矩阵。

$$\mathbf{A}=\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\end{bmatrix}\qquad \mathbf{A}^{\mathrm{T}}=\begin{bmatrix}1 & 4\\2 & 5\\3 & 6\end{bmatrix}$$

图 7-5

7.1.6 矩阵的秩

矩阵的秩是一个标量，在矩阵代数的某些方面有作用。只有方阵才有秩。矩阵 **A** 的秩用|**A**|或 det(**A**) 表示。求 2×2 或 3×3 矩阵的秩相当容易。

下面是一个 2×2 的矩阵(见式(7-9))：

$$\mathbf{A}=\begin{bmatrix}a_{11} & a_{12}\\a_{21} & a_{22}\end{bmatrix}\tag{7-9}$$

它的秩的计算公式如式(7-10)所示：

$$|\mathbf{A}|=a_{11}a_{22}-a_{21}a_{12}\tag{7-10}$$

式(7-11)中的 B 是一个 3×3 矩阵：

$$\mathbf{B}=\begin{bmatrix}b_{11} & b_{12} & b_{13}\\b_{21} & b_{22} & b_{23}\\b_{31} & b_{32} & b_{33}\end{bmatrix}\tag{7-11}$$

它的秩计算公式如下式(7-12)所示：

$$|\mathbf{B}|=b_{11}\begin{vmatrix}b_{11} & b_{12}\\b_{21} & b_{22}\end{vmatrix}-b_{12}\begin{vmatrix}b_{21} & b_{23}\\b_{31} & b_{33}\end{vmatrix}+b_{13}\begin{vmatrix}b_{21} & b_{22}\\b_{31} & b_{32}\end{vmatrix}\tag{7-12}$$

按同样方式求阶数比较大的矩阵的秩时，比起手工计算，用 Excel 和 MATLAB 计算矩阵的秩就容易得多。

7.1.7 矩阵的逆

一个矩阵的逆具有这样的性质：它与原来的矩阵相乘得到一个单位矩阵。这个性质可以用下面的式子来表示：

$$\mathbf{A}\mathbf{A}^{-1}=\mathbf{I} \tag{7-13}$$

A 矩阵的逆矩阵用 $\mathbf{A}^{-1}$ 表示。一个矩阵与它的逆矩阵相乘是可以交换的，即 $\mathbf{A}\mathbf{A}^{-1}=\mathbf{A}^{-1}\mathbf{A}=\mathbf{I}$ 并非所有的矩阵都有逆矩阵，只有方阵才有逆矩阵。求矩阵的逆矩阵非常繁琐，它的求法留到线性代数课上介绍。我们将利用 Excel 和 MATLAB 的强大功能求矩阵的逆。

7.2 教程：Excel 里的矩阵运算

Excel 内置许多矩阵运算的函数。本教程将向读者介绍以下内容：

- 在 Excel 里矩阵的表示方法。
- 如何在 Excel 里进行矩阵相加运算。
- 如何在 Excel 里进行矩阵相乘运算。
- 如何在 Excel 里进行矩阵与标量相乘运算。
- 在 Excel 里转置矩阵。
- 在 Excel 里求矩阵的秩。
- 在 Excel 里求矩阵的逆。

7.2.1 在 Excel 里矩阵的表示和相加运算

矩阵的每个元素必须输入到每个单元里。如图 7-6 所示，在 B2:D4 单元格里输入数据。并把单元格式设置为居中对齐。

	1	4	1
A=	1	4	5
	3	2	7

图 7-6

在 A3 单元里输入 A，表示它是这个矩阵的名称。然后，我们用粗边框当作矩阵的方括号，说明它是一个矩阵。单击 Ribbon 字体组里的边框下拉菜单，从【绘制边框】菜单中选择线型，如图 7-7 所示。选择了合适的线型后，垂直拖动画线工具，画竖直线，如图 7-8 所示。按 Esc 键，关闭画线工具。在 Excel 里，添加垂直括号线只是为了外表好看，并不是必需的。

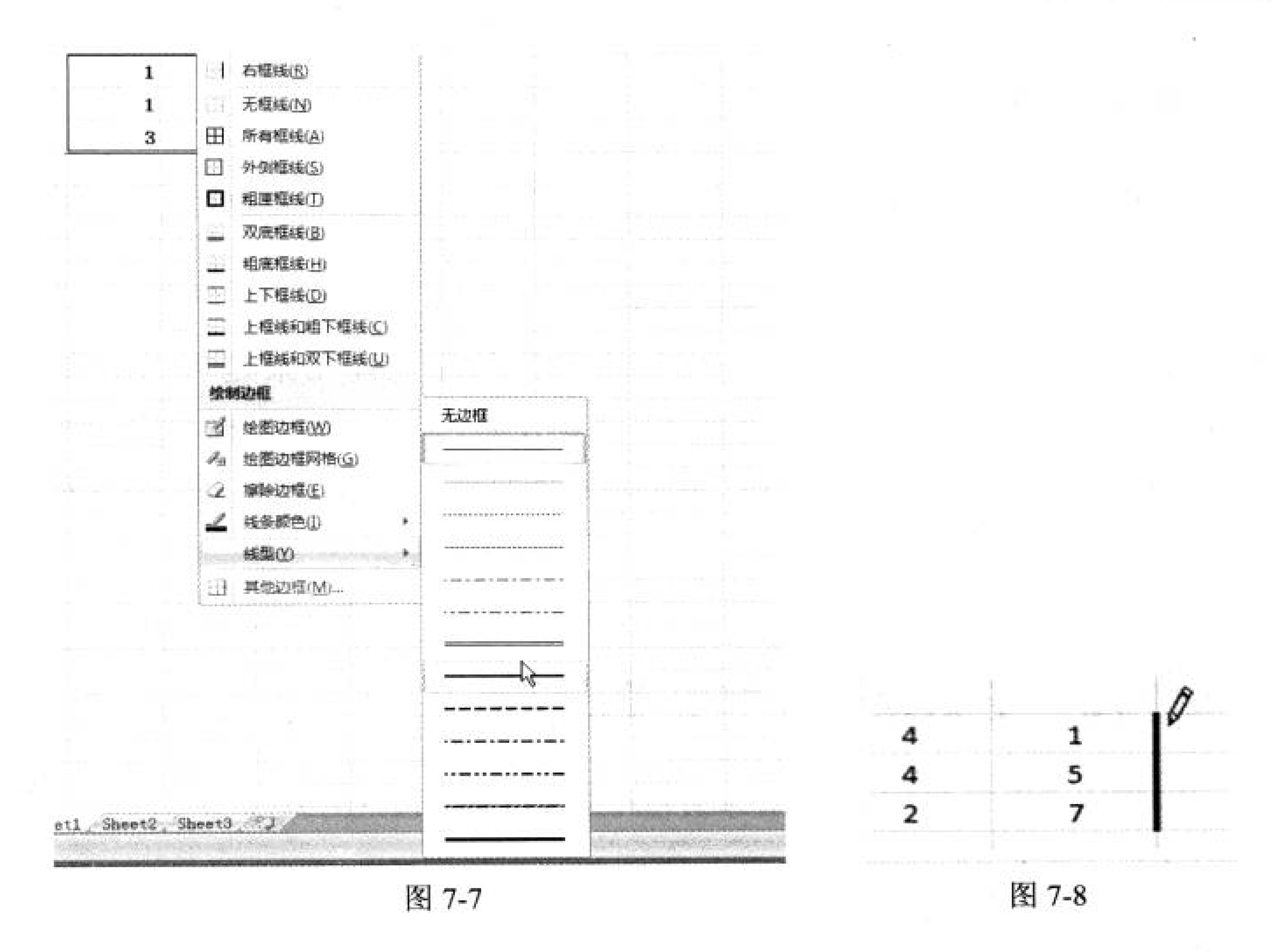


图 7-7　　　　图 7-8

现在，把图 7-9 的矩阵输入到 G2:I4 单元格里，并且居中对齐。用同样的方法在 F3 单元格里输入矩阵名称 B 并设置矩阵括号。

当我们用 Excel 进行运算时，必须知道有关矩阵阶数的一些规定。这是因为我们必须事先确定好运算结果矩阵的阶数。即要选好结果矩阵所占用的单元区域。例如，当两个 3×3 矩阵相加，必须选好 3×3 单元区域存放运算结果。选择好保存运算结果的单元区域后，我们在公式里输入矩阵运算的公式，必须按住 Ctrl+Shift 组合键后按 Enter 键才会执行矩阵运算。

现在我们把 **A** 矩阵与 **B** 矩阵相加，得到一个 **C** 矩阵。选择单元区域 L2:N4，在公式栏里输入“＝B2:D4+G2:I4”。需要提醒读者的是，直接选择矩阵 **A** 和 **B** 的区域比输入单元格地址要快许多。按 Ctrl+Shift+Enter 组合键，告诉 Excel 这是一个矩阵运算。Excel 则把这个公式看成矩阵运算，运行此运算，并把结果填入 L2:N4 的相应单元格里。Excel 为了表示此公式是数组或矩阵运算，自动在公式的两端加上一对花括号，如图 7-10 所示。

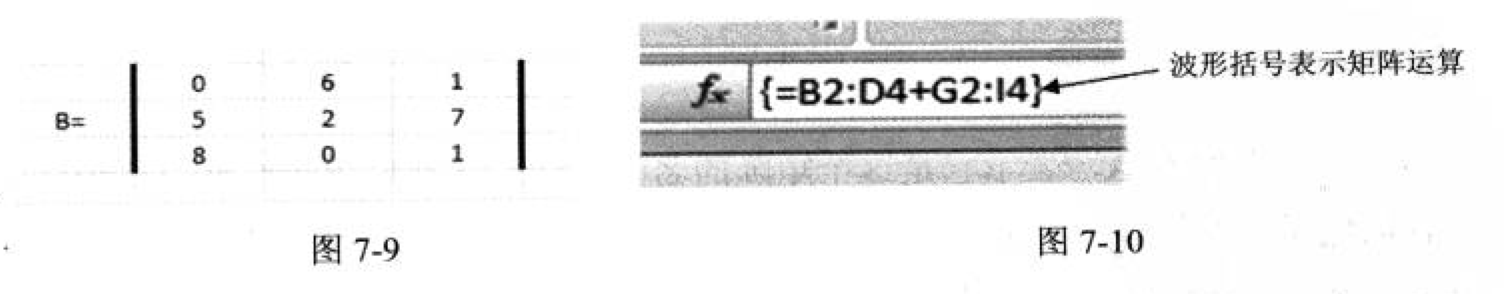


图 7-9　　　　图 7-10

在 K3 单元格里输入“A＋B=”文字，并在和矩阵的两边加上两条竖直线，再把单元格设置为居中对齐格式。最后得到的电子表格如图 7-11 所示。需要说明的是，现在我们在 A 和 B 中输入任意值，Excel 就会在单元 L2:N4 立刻显示两个矩阵的和。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		1	4	1			0	6	1			1	10	2
3	A=	1	4	5		B=	5	2	7		A+B=	6	6	12
4		3	2	7			8	0	1			11	2	8
5														

图 7-11

7.2.2 Excel 里的矩阵相乘和转置运算

如图 7-12 所示，输入标签并设置矩阵 **A** 和 **B** 的格式。

接着执行 **C**=**AB** 运算。两个矩阵相乘，它们的阶数必须符合要求。现在我们把一个 3×3 矩阵乘上一个 3×2 矩阵，根据图 7-3，它们的内部阶数必须一致，才可以进行相乘运算，它们的两个外部阶数(3×2)决定了积矩阵的阶数。

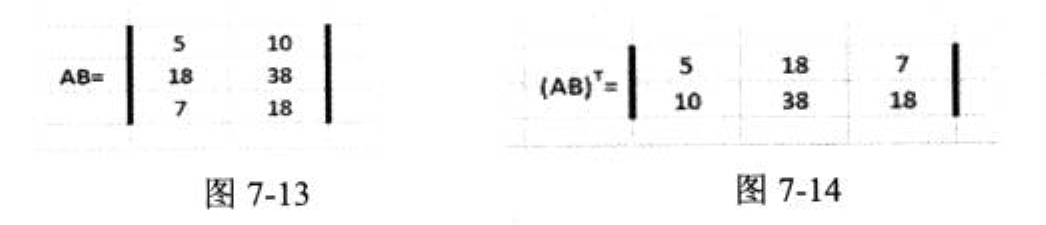
选择 B6:C8 单元格，作为积矩阵的存放单元区域。在公式栏里输入“=MULTI (B2:D4,G2:H4)”公式，然后按 Ctrl+Shift+Enter 组合键，结束公式输入，在单元 A7 里输入标签“AB=”，然后设置这个积矩阵的格式，如图 7-13 所示。

	A	B	C	D	E	F	G	H
1								
2		1	2	0			1	6
3	A=	3	5	5		B=	2	2
4		2	2	1			1	2
5								

图 7-12

下面我们转置这个刚生成的矩阵。同样要确定 **C**=(**AB**)^T 这个积矩阵的阶数，矩阵 **AB** 的阶数为 3×2，因此，它的转置矩阵阶数为 2×3，选择 G6:I7 单元格，然后在公式栏里输入“=TRANSPOSE(B6:C8)”，按 Ctrl+Shift+Enter 组合键结束输入。

合并 F6:F7 单元格，输入“(AB)^T=”标签，正确设置转置矩阵的格式，把它设置为如图 7-14 所示。



AB=		
	5	10
	18	38
	7	18

(AB)^T=			
	5	18	7
	10	38	18

图 7-13　　　　图 7-14

最后，我们进行标量与矩阵相乘运算。在 A11 单元格里输入“k=”，在 B11 单元格里输入 3。在这个例子里，我们把矩阵 **B** 乘上标量 k。矩阵乘上标量不会改变阶数，因此选择一个 3×2 区域存放结果矩阵。选择 G10:H12，然后在公式栏里输入“=B11*G2:H4”公式，按 Ctrl+Shift+Enter 组合键，结束输入。

在 F11 单元格里输入“kB=”，然后给新矩阵设置合适的格式，最后得到的电子表格如图 7-15 所示。

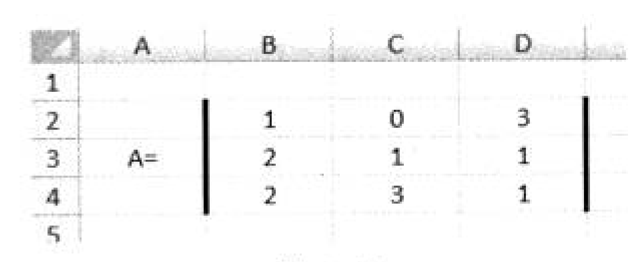
	A	B	C	D	E	F	G	H	I
1									
2		1	2	0			1	6	
3	A=	3	5	5		B=	2	2	
4		2	2	1			1	2	
5									
6		5	10				5	18	7
7	AB=	18	38			$(AB)^T$=	10	38	18
8		7	18						
9									
10							3	18	
11	k=	3				kB=	6	6	
12							3	6	

图 7-15

7.2.3　在 Excel 里求矩阵的秩和逆

打开一个新电子表格，按图 7-16 的要求，输入矩阵符号，并且设置矩阵的格式。

现在我们计算矩阵 **A** 的秩。在单元 F3 里输入“det(A)”，然后在单元 G3 里，输入公式“=MDETERM(B2:D4)”，读者的电子表格应该如图 7-17 所示。



	A	B	C	D
1				
2		1	0	3
3	A=	2	1	1
4		2	3	1
5				

图 7-16

	A	B	C	D	E	F	G
1							
2		1	0	3			
3	A=	2	1	1		det(A)=	10
4		2	3	1			
5							

图 7-17

记住，只有方阵才有逆矩阵。与其他矩阵运算一样，我们必须知道结果矩阵的阶数。在本例中，$\mathbf{C}=\mathbf{A}^{-1}$，结果矩阵的阶数与原矩阵 **A** 的阶数一样。选择 B6:D8 单元格，然后在公式栏里输入公式“=MINVERSE(B2:D4)”，再按 Ctrl+Shift+Enter 组合键，结束操作。

我们用 $\mathbf{A}^{-1}$ 表示这个矩阵，为此在单元 A7 输入“A^{-1}”，再在两边加上粗线表示方括号，把数据对齐格式设为居中，把小数位数设置为 2 位。得到的电子表格如图 7-18 所示。

	-0.2	0.9	-0.3
A^{-1}=	0	-0.5	0.5
	0.4	-0.3	0.1

图 7-18

如果在 B2:D4 单元格里输入一个新的 3×3 矩阵，则在单元格 B6:D8 出现它的逆矩阵。

现在我们修改矩阵 **A** 的一个元素。把 a_{32} 的值从 3 改为 1，则电子表格的结果如图 7-19 所示。

	A	B	C	D	E	F	G
1							
2		1	0	3			
3	A=	2	1	1		det(A)=	0
4		2	1	1			
5							
6		#NUM!	#NUM!	#NUM!			
7	A⁻¹=	#NUM!	#NUM!	#NUM!			
8		#NUM!	#NUM!	#NUM!			

图 7-19

在逆矩阵里出现了许多“!NUM”，这表示无法求得矩阵 **A** 的逆矩阵。记住，并不是所有的方阵都存在逆矩阵。在我们这个矩阵里，第 2 行与第 3 行的元素相同。这表示，这两行是线性相关的。矩阵可逆的一个条件是，行与行之间是线性独立的。除了非线性独立的矩阵外，某一行的元素都为 0 的矩阵也是不可逆的。从图上还可以看出，此时的秩也为 0。秩为 0 的矩阵是奇异阵，计算一个矩阵的秩，如果它不为 0，则它存在可逆矩阵，这是判断可逆矩阵的一种很有用的方法。

用撤消工具，把矩阵 **A** 恢复到原来的值。

现在我们来验证 $\mathbf{AA}^{-1}$ 是否等于 **I**。选择 B10:D12 单元区域，在公式栏里输入公式“=MMULT(B6:D8,B2:D4)”，然后按 Ctrl+Shift+Enter 组合键，在单元 A11 里输入“$\mathrm{AA}^{-1=}$”，再在两边加上粗线，最后得到的矩阵应该与图 7-20 类似。

	1	1.1E-16	-5.6E-17
A⁻¹A=	0	1	0
	5.6E-17	5.6E-17	1

图 7-20

得到的乘积矩阵看起来不像是一个单位矩阵。在非对角线位置上的一些元素出现非 0 值。但是仔细观察，发现这些值都非常小，这是因为在求逆运算中存在数值误差的原因。这些值超出了运算的精度之外，因此，它们完全可以忽略不计。为了使矩阵看起来整齐，我们把小数位数设置为 2，并把对齐格式设置为居中。最后得到的电子表格如图 7-21 所示。

	A	B	C	D	E	F	G
1							
2		1	0	3			
3	A=	2	1	1		det(A)=	10
4		2	3	1			
5							
6		-0.20	0.90	-0.30			
7	A⁻¹=	0.00	-0.50	0.50			
8		0.40	-0.30	0.10			
9							
10		1.00	0.00	0.00			
11	A⁻¹A=	0.00	1.00	0.00			
12		0.00	0.00	1.00			

图 7-21

7.3　教程：MATLAB 里的矩阵运算

MATLAB 有许多内置的矩阵运算函数。本教程将向读者介绍以下内容：

- 在 MATLAB 里正确输入矩阵。
- 在 MATLAB 里执行矩阵相加运算。
- 在 MATLAB 里执行矩阵相乘运算。
- 在 MATLAB 里执行矩阵与标量相乘运算。
- 在 MATLAB 里求矩阵的转置。
- 在 MATLAB 里求矩阵的秩。
- 在 MATLAB 里求矩阵的逆矩阵。

7.3.1　MATLAB 里的矩阵排列和相加运算

MATLAB 里的矩阵相加运算用“+”运算符表示。要对矩阵 **A** 和 **B** 进行相加运算，我们必须先建立变量 **A** 和 **B**，并给每个矩阵赋值。回忆一下第 3 章学过的内容，建立一个矩阵，需要输入方括号，元素之间用空格或逗号分隔，用分号表示另一起行。下面的命令建立矩阵 **A** 和 **B**：

```
>> A=[1 0 2; 2 1 2; 0 2 1]
A =
     1 0 2
     2 1 2
     0 2 1

>> B=[1 0 1; 1 1 2; 3 3 1]
B =
     1 0 1
     1 1 2
     3 3 1
```

下面的命令求矩阵 **C**＝**A**+**B**：

```
>> C=A+B
C =
    2 0 3
    3 2 4
    3 5 2
```

7.3.2　用 MATLAB 进行矩阵相乘运算

矩阵的相乘运算用“*”运算符表示。下面的命令利用第 7.3.1 节定义的矩阵 **A** 和 **B**，求 **C**＝**AB**：

```
>> C=A+B
C =
      7 6 3
      9 7 6
      5 5 5
```

矩阵与标量相乘运算也可用“*”运算符。下面这个命令把矩阵 **C** 乘上 2 再赋给 **C**，实际上，就是将 **C** 的每个元素都放大两倍：

```
>> C=2*C
C =
      2  0 4
      4  2 8
      0 12 2
```

7.3.3 用 MATLAB 求转置矩阵

在 MATLAB 求转置矩阵需要用“'”运算(单撇号)，为了说明它的用法，我们先定义矩阵 **A** 和 **B**，如下所示：

```
>> A=[0 1 1; 2 3 1; 1 2 1]
A =
    0 1 1
    2 3 1
    1 2 1
>> B=[1 2 2; 1 0 1]
B =
    1 2 2
    1 0 1
```

现在求 **C**=**AB**，命令如下：

```
>> C=A*B
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

前面曾说过，两个矩阵相乘，第一个矩阵的列数必须等于第二个矩阵的行数。这里，MATLAB 返回一个错误信息，告诉我们由于两个矩阵的阶数不符合乘法要求，因此不能相乘。

有时，我们输入矩阵时，行与列搞反了，这时，求它的转置矩阵很快能把它恢复正确，这比重新输入要快许多，特别当矩阵很大时，尤其如此。另一种情况是某个运算得到一个矩阵，这个矩阵必须转置后才可能参加后面的运算。现在把 **B** 矩阵转置过来，再执行 **C**=**AB** 运算，命令如下：

```
>> B=B'
B =
      1 1
      2 0
```

```
    2 1

>> C=A*B
C =
    4 1
    10 3
    7 2
```

7.3.4　用 MATLAB 求逆矩阵

在 MATLAB 里，求逆矩阵用 inv 命令，而且只有方阵才可以求逆矩阵。现在我们重新定义矩阵 **A** 和 **B**，如下所示：

```
>> A=[1 2 1; 0 2 0; 2 1 1]
A =
    1 2 1
    0 2 0
    2 1 1

>> B=[1 2 3; 4 5 6; 7 8 9]
B =
    1 2 3
    4 5 6
    7 8 9
```

现在执行 $\mathbf{C}=\mathbf{A}^{-1}$ 和 $\mathbf{D}=\mathbf{B}^{-1}$ 运算，命令如下：

```
>> C=inv(A)
C =
   -1.0000    0.5000     1.0000
         0    0.5000     0
    2.0000   -1.5000    -1.0000

>> D=inv(B)
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND =
         1.541976e-018.
D =
   1.0e+016 *
  -0.4504     0.9007    -0.4504
   0.9007    -1.8014     0.9007
  -0.4504     0.9007    -0.4504
```

这里，MATLAB 发出一个警告消息，表示矩阵 **B** 非常接近于奇异矩阵。虽然在屏幕输出 **D** 矩阵，但是要注意矩阵前面的比例因子，即 **D** 的每个元素都要乘 10^{16}。因此 **D** 不能用在之后的运算中。

还记得前面的Excel例子吗？矩阵的秩会告诉我们一个矩阵是否可逆。如果秩等于0(奇异矩阵的定义)，则矩阵肯定不可逆，相反，秩不为 0，则肯定存在逆矩阵。

现在我们通过求 **A** 和 **B** 的秩，验证这个结论。输入以下命令：

```
>> detA=det(A)
detA =
    -2

>> detB=det(B)
detB =
     0
```

矩阵运算不必每次只执行一个运算，我们可以在一行内执行多个运算。例如，下面的命令先定义两个矩阵 **A** 和 **B**，然后用一个命令执行 $\mathbf{C}=(\mathbf{AB}')^{-1}$ 运算。

```
>> A=[1 2 1; 0 1 2]
A =
     1  2  1
     0  1  2

>> B=[1 2 0; 1 1 2]
B =
     1  2  0
     1  1  2

>> C=inv(A*B')
C =
     0.3333  -0.3333
    -0.1333   0.3333
```

7.4 习题

习题 1~习题 8 中的 A 和 B 矩阵如下：

$$\mathbf{A}=\begin{bmatrix}1 & 3 & 0\\2 & 1 & 2\\4 & 1 & 3\end{bmatrix} \qquad \mathbf{B}=\begin{bmatrix}0 & 1 & 1\\1 & 3 & 3\\2 & 0 & 3\end{bmatrix}$$

1. 用 Excel 求 **C**＝**A**＋**B**。用手工求 **C**＝**A**＋**B**，验证 Excel 的计算结果。
2. 用 Excel 证明对于上面的矩阵 **A** 和 **B**，存在 $\mathbf{AB}\neq\mathbf{BA}$ 关系。
3. 用 Excel 求矩阵 **A** 和 **B** 的秩。每个矩阵都有一个逆矩阵吗？利用 **A** 和 **B** 的秩验证你的结论。
4. 用 Excel 证明 $\mathbf{AA}^{-1}=\mathbf{A}^{-1}\mathbf{A}$。
5. 用 MATLAB 求 **C=A+B** 的值。用手工方法求 **C=A+B**，验证 MATLAB 的运算结果。
6. 用 MATLAB 证明对于上面的矩阵 **A** 和 **B**，存在 $\mathbf{AB}\neq\mathbf{BA}$。
7. 用 MATLAB 求矩阵 **A** 和 **B** 的秩。每个矩阵都有一个逆矩阵吗？利用 **A** 和 **B** 的秩验证你的结论。

8. 用 MATLAB 证明 $\mathbf{AA}^{-1}=\mathbf{A}^{-1}\mathbf{A}$。

习题 9 和习题 10 用到的 **D** 矩阵如下：

$$\mathbf{D}=\begin{bmatrix}12 & -6 & 27\\ 5 & 18 & -2\\ -4 & 2 & -9\end{bmatrix}$$

9. 用 Excel 求矩阵 **D** 的秩和逆矩阵。

10. 用 MATLAB 求矩阵 **D** 的秩和逆矩阵。

习题 11~习题 14 用到的矩阵 **G** 和 **H** 取如下值：

$$\mathbf{G}=\begin{bmatrix}1 & 1 & 0\\ 0 & 1 & 2\\ 2 & 2 & 3\end{bmatrix} \qquad \mathbf{H}=\begin{bmatrix}1 & 3 & 4\end{bmatrix}$$

11. 对于上面的 **G**，用 Excel 证明 $(\mathbf{G}^{\mathrm{T}})^{\mathrm{T}}=\mathbf{G}$。

12. 对于上面的 **G** 和 H，用 Excel 求 $\mathbf{C}=\mathbf{GH}^{\mathrm{T}}$。

13. 对于上面的 **G**，用 MATLAB 证明 $(\mathbf{G}^{\mathrm{T}})^{\mathrm{T}}=\mathbf{G}$。

14. 对于上面的 **G** 和 **H**，用 MATLAB 求 $\mathbf{C}=\mathbf{GH}^{\mathrm{T}}$。

习题 15 和习题 16 用到矩阵 **M** 和 **N**，它们的值如下：

$$\mathbf{M}=\begin{bmatrix}1 & 3 & 3 & 6 & 4\\ 2 & 1 & 4 & 2 & 4\\ 6 & 4 & 1 & 3 & 8\\ 0 & 4 & 6 & 2 & 1\\ 4 & 2 & 0 & 4 & 1\end{bmatrix} \qquad \mathbf{N}=\begin{bmatrix}20\\ 6\\ 14\\ 9\\ 5\end{bmatrix}$$

15. 用 Excel 求 $\mathbf{X}=\mathbf{M}^{-1}\mathbf{N}$ 的值，**M** 和 **N** 取上述值。

16. 用 MATLAB 求 $\mathbf{X}=\mathbf{M}^{-1}\mathbf{N}$ 的值，**M** 和 **N** 取上述值。

习题 17 和习题 18 用到 **P** 和 **Q** 矩阵如下：

$$\mathbf{P}=\begin{bmatrix}1 & 3 & 3 & 6 & 4\\ 2 & 1 & 4 & 2 & 4\\ 6 & 4 & 1 & 3 & 8\\ 0 & 4 & 6 & 2 & 1\\ 4 & 2 & 0 & 4 & 1\end{bmatrix} \qquad \mathbf{Q}=\begin{bmatrix}10 & 12 & 14 & 7 & 3\end{bmatrix}$$

17. 用 Excel 求 $\mathbf{X}=\mathbf{P}^{-1}\mathbf{Q}^{\mathrm{T}}$ 的值，**P** 和 **Q** 的值如上。

18. 用 MATLAB 求 $\mathbf{X}=\mathbf{P}^{-1}\mathbf{Q}^{\mathrm{T}}$ 的值，**P** 和 **Q** 的值如上。

习题 19 和习题 20 用到的 **R** 和 **S** 矩阵如下：

$$\mathbf{R}=\begin{bmatrix}1 & 0 & 0 & 4 & 1\\ 2 & -2 & 0 & 1 & -1\\ 0 & 4 & 1 & -3 & 0\\ 1 & -1 & 2 & 2 & 1\\ 3 & 2 & 5 & 0 & 1\end{bmatrix} \qquad \mathbf{S}=\begin{bmatrix}11\\ -7\\ 3\\ -17\\ 5\end{bmatrix}$$

19. 用 Excel 求 $\mathbf{Z}=\mathbf{R}^{-1}\mathbf{S}$ 的值，**R** 和 **S** 矩阵的值如上。
20. 用 MATLAB 求 $\mathbf{Z}=\mathbf{R}^{-1}\mathbf{S}$ 的值，**R** 和 **S** 矩阵的值如上。

第 8 章 求方程组的根

引言

在许多实际工程问题中，必须同时求解多个方程的根，即求得的解必须满足所有的方程。有时为了解决一个工程问题，我们需要按顺序求解两个或多个方程。但是在复杂的工程分析中，甚至需要求解成千上万个方程。

本章我们要学习以下内容：

- 如何用矩阵表示一个线性方程组。
- 如何在 MATLAB 和 Excel 里用矩阵运算求解线性方程组。
- 如何在 Excel 里用规划求解工具求解一组非线性方程。

8.1 线性方程组

在分析工程问题时，通常会得到一个需要同时求解的方程组，求解的方法取决于方程组中各个方程的类型。通常，这些方程都是线性的，即方程组中只包含常量或常量与一次方变量的乘积。在第 6 章曾讨论过线性方程这个概念。假如 x 是方程中的一个变量，则它只能以 x，$3x$，$6x$ 或者其他常量与 x 的乘积出现，而不能以 $x^2,\sqrt{x}$,或者其他更复杂的函数的形式出现(回忆一下，非线性方程才可以包含这些非线性项)。在工程问题中出现的线性方程可能会包含很多变量，但是我们先考虑以下两个方程：

$$2x+3y=14 \tag{8-1}$$

$$-4x+y=28 \tag{8-2}$$

为了解这一组方程，方程的个数必须与未知量的个数相等。要使一组解(x 和 y)同时满足方程(8-1)和(8-2)，则这两个方程必须符合以下条件：(1)独立性，即一个方程不能是另一个方程与一个常量的乘积；(2)一致性，两个方程不能只是在常量项上有差别。例如，下面

的方程(8-3)和(8-4)不是两个独立方程，因为其中一个方程乘以一个常量，就可得出另一个方程，实际上它们代表同一个方程。

$$2x+3y=14 \tag{8-3}$$

$$4x+6y=28 \tag{8-4}$$

方程(8-5)和(8-6)存在不一致性，因为 x 和 y 分别乘以 2 和 3，得到的结果却是 14 和 28，这是不可能的。这两个方程实际上代表两条平行直线，不可能相交，即不可能存在一个解同时满足这两个方程。

$$2x+3y=14 \tag{8-5}$$

$$2x+3y=28 \tag{8-6}$$

下面我们先用 Excel 的图形法，然后用单变量求解方法求解一组线性方程。最后通过 Excel 和 MATLAB 里的矩阵运算求解方程组。非线性方程组的求解使用 Excel 的规划求解工具。

8.2　教程：用 Excel 求解线性方程组

例 8.1

用 Excel 图形法求解(8-1)和(8-2)线性方程组的解。我们在此重写此方程组：

$$2x+3y=14 \tag{8-7}$$

$$-4x+y=28 \tag{8-8}$$

解：

它们都是直线方程。为了用图形法确定它们的解，我们必须把它们改写为斜率——截距的表示形式，即 y=mx+b 的形式。把这两个方程含有 y 的项移动到方程的左边。方程(8-1)经移项后得到方程(8-9)，两边除以 y 前面的系数，得到方程(8-10)。把方程(8-2)化简后，得到方程(8-11)。

$$3y=-2x+14 \tag{8-9}$$

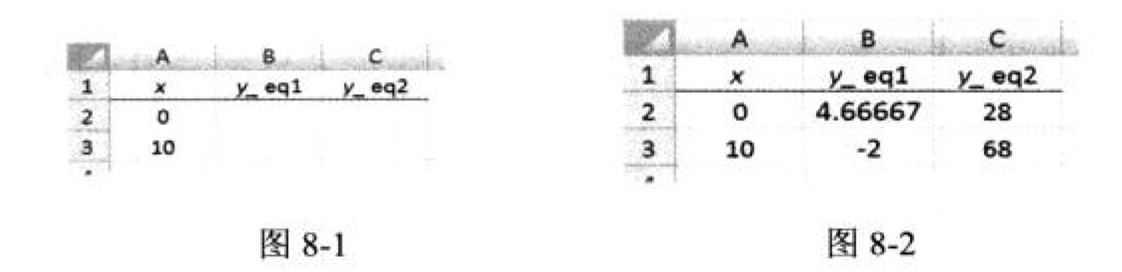
$$y=-\frac{2}{3}x+\frac{14}{3} \tag{8-10}$$

$$y=4x+28 \tag{8-11}$$

在 Excel 中打开一个空白的工作表。在 A1 单元格里输入 x 文字标签，表示方程(8-10)和(8-11)的自变量 x 的值。在单元格 B1 和 C1 里分别输入 y_eq1 和 y_eq2，表示方程(8-10)和(8-11)的 y 值。在 A2 和 A3 单元格分别输入 0 和 10。由于它们是直线方程，而且只需要固定两个点就可以画出它们的直线，因此，x 可以取任意两个值。按图 8-1 的样式设置电子表格格式。

在单元 B2 里，输入公式“=-2/3*A2+14/3”，表示方程(8-10)的 y 值。然后把它复制到单元 B3。

在单元 C2 里，输入公式“=4*A2+28”，表示方程(8-10)的 y 值。然后把它复制到单元 C3。最后得到的电子表格如图 8-2 所示。



	A	B	C
1	x	y_eq1	y_eq2
2	0		
3	10		

图 8-1

	A	B	C
1	x	y_eq1	y_eq2
2	0	4.66667	28
3	10	-2	68

图 8-2

现在用带直线的散点图表示这两个方程。添加垂直轴的主网格线，并把图形移动到另一个空白的工作表上。经过格式设置后，得到如图 8-3 所示的图形。

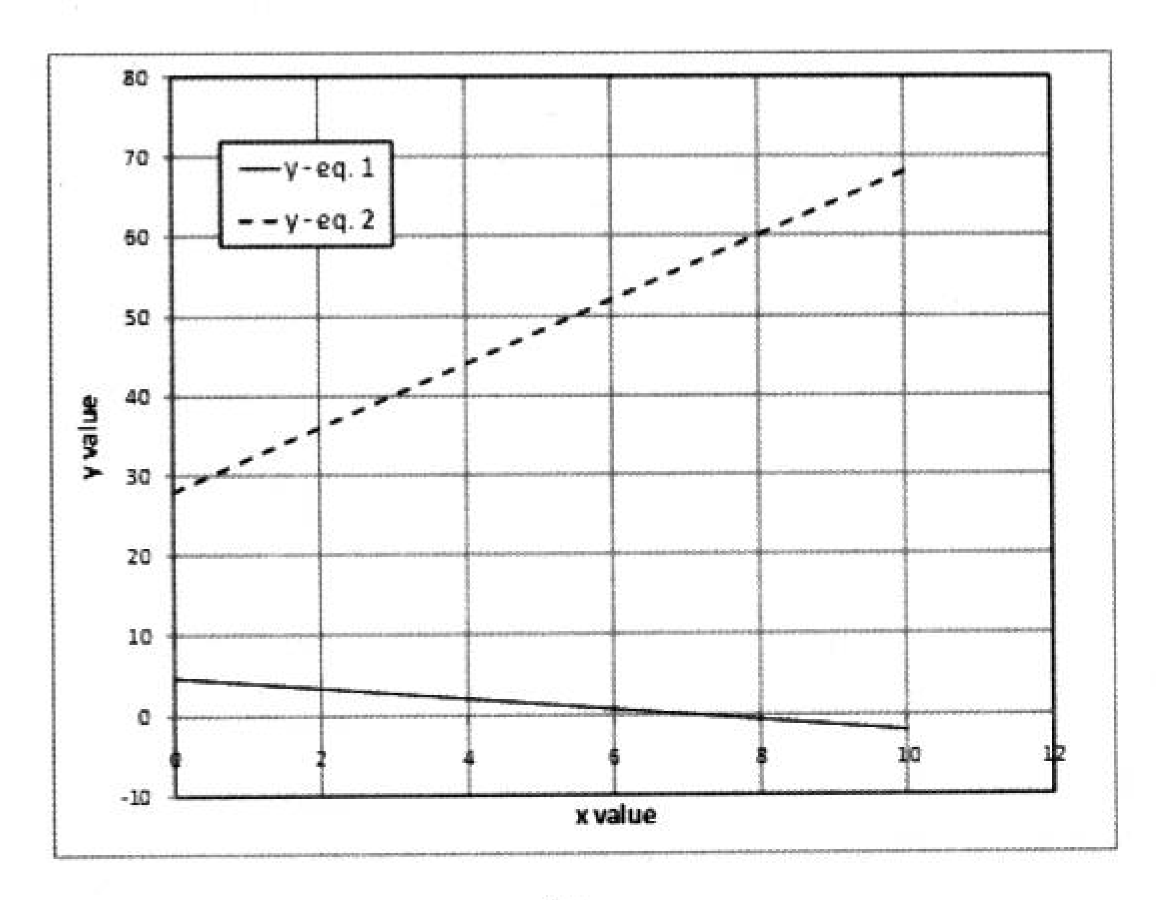


图 8-3

这两个方程的解是满足此方程组的唯一解。它就是这两条直线的交点，但是在这个图上看不到它们的交点。从图 8-3 我们看出，这两条直线的交点大概在-10~0 之间。改变横坐标轴的刻度，重绘这个图形。把 A3 单元里的值改为-10。现在得到如图 8-4 所示的图形。现在这两条直线相交，我们发现交点在 x=-5 附近。但是，由于受图形的分辨率限制，我们还不能确定它就是这个方程组的解。

现在我们增大的图形的分辨率。在 A2 和 A3 单元里分别输入-4 和-6，更新绘制图形，得到如图 8-5 所示的图形。从图 8-5 可以看出，方程的解为(-5,8)。这种图形法是一种非常有效的求解方法。但是，当解为非整数时，很难得到一个准确值，因此最好采用下面介绍的方法。

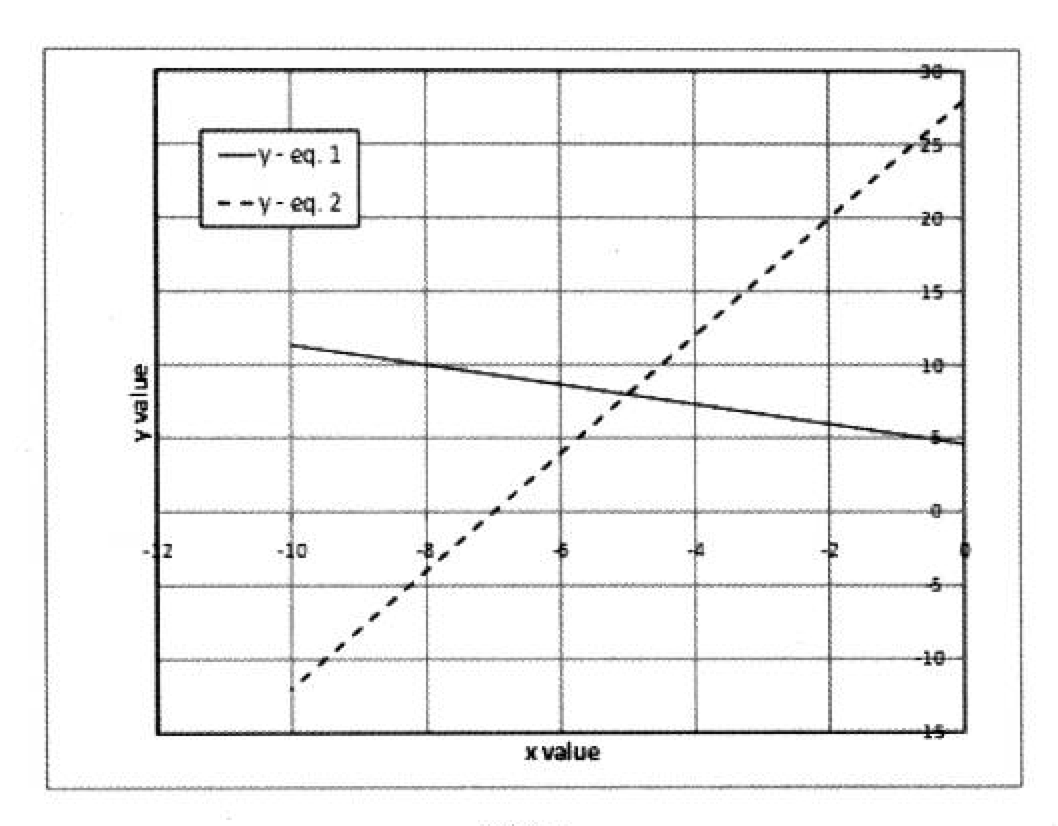


图 8-4

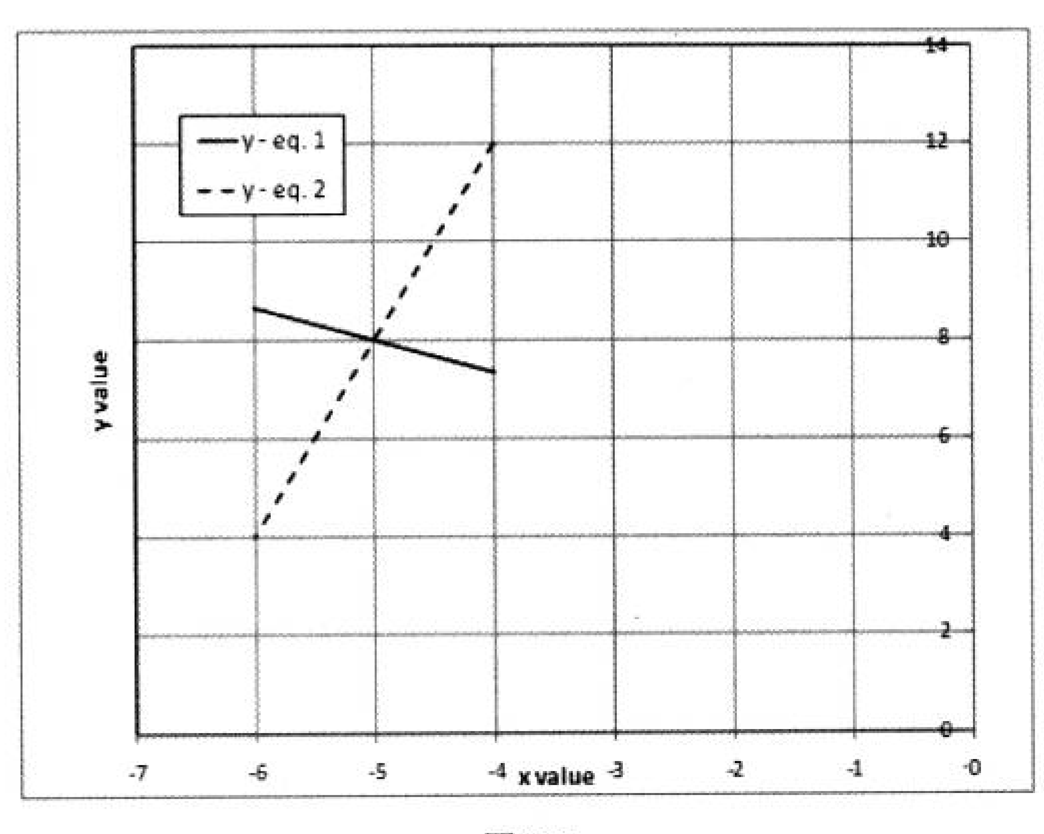


图 8-5

例 8.2

用 Excel 的单元变量求解法解例 8.1 的方程组。

解：

在 Excel 里，求解方程的第二种方法是利用 Excel 的单变量求解工具。回忆一下第 6 章中介绍的内容。单变量求解方法通过设置第二个单元的值可以自动改变第一个单元的值。当只有两个未知变量时，单变量求解方法可以用来解方程组。

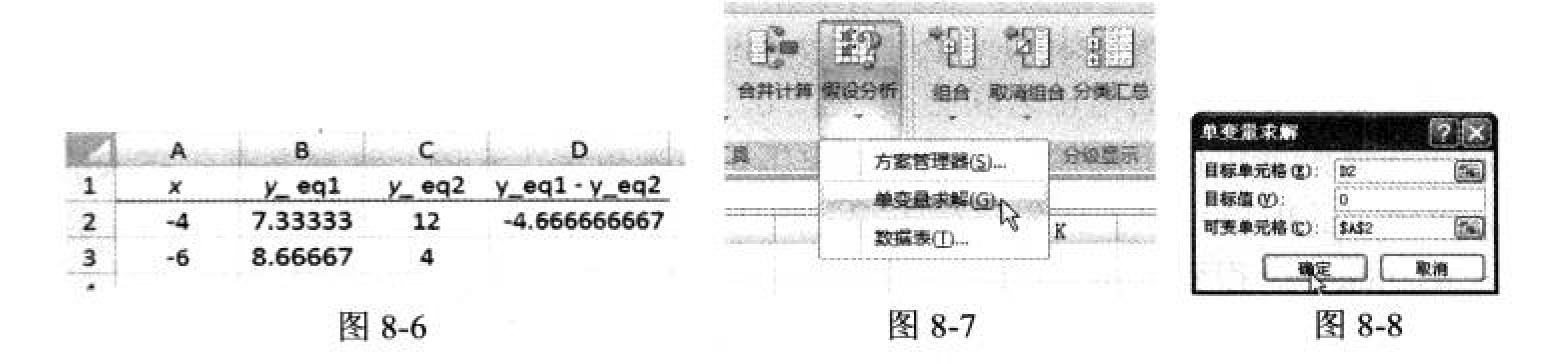
现在我们使用前面例子的电子表格。通过改变第二个变量的值，让一个变量趋于一个已知值。在本例中，我们取两个方程的 y 值之差，通过改变 x 值，使它趋向于 0。当它们之差为 0 时，它们的值相等，方程的解也就找到了。

在 D1 中输入“y_eq1-y_eq2”，在 D2 中输入公式“=B2-C2”。正确设置单元的格式，使它与图 8-6 相似。

现在我们用单变量求解工具求方程的解。选择单元 D2，然后从 Ribbon 中选择数据选项卡，再从【假设分析】的下拉列表中选择【单变量求解】，如图 8-7 所示。

接着出现一个【单变量求解】的对话框，把【目标单元格】设置为 D2，把目标值设置为 0，把【可变单元格】设置为 A2，如图 8-8 所示，然后单击【确定】按钮，关闭单变量求解对话框。

这样我们将得到方程组的另一个解。



	A	B	C	D
1	x	y_eq1	y_eq2	y_eq1 - y_eq2
2	-4	7.33333	12	-4.666666667
3	-6	8.66667	4	

图 8-6　　图 8-7　　图 8-8

例 8.3

用 Excel 矩阵运算解下面的电路分析问题。

在这个例子里，我们用 Excel 求解一个非常简单的方程组。在下面的练习里，我们将充分利用 MATLAB 的编程功能，求解大型的方程组。

电子设计工程师们经常在电阻电路分析中遇到线性方程组。现在我们就以电路分析为例介绍方程组的求解方法。首先，我们来分析一个像图 8-9 那样非常简单的电路。图中电流的单位用 A 表示，电路中的电源提供了电流，电流的大小由连接到电池上的电阻或负载决定。如果只有一个电阻连接到电源，则流过此电路的电流由安培定律，即方程(8-12)决定：

$$I = \frac{\Delta V}{R} \tag{8-12}$$

式(8-12)中，ΔV 是加在电阻上的电压(本例中，它就等于电压源的电压)，单位为伏特(V).I 是流过电阻的电流(也是流过此电路其他部分的电流)，单位为 A。R 是电阻的大小，单位为 Ω。

假如有一个 12V 的电池，连接到 2Ω 的电阻上，通过这个电阻的电流为 $12\text{V}/2\Omega=6\text{A}$。

现在我们来分析一个稍微复杂一些的电路，如图 8-10 所示，在这个电路里，为两个并联电阻电路提供电流。在电路网络里流过某个结点的电流总和必须等于 0。换句话说，流入结点的电流必须等于从该结点流出的电流。这就是基尔霍夫的电流定律。我们把这个定律应用到这个网络的 $R1$、$R2$ 和 $R3$ 的连接结上，则得到下面的方程：

$$I_1 - I_2 - I_3 = 0 \tag{8-13}$$

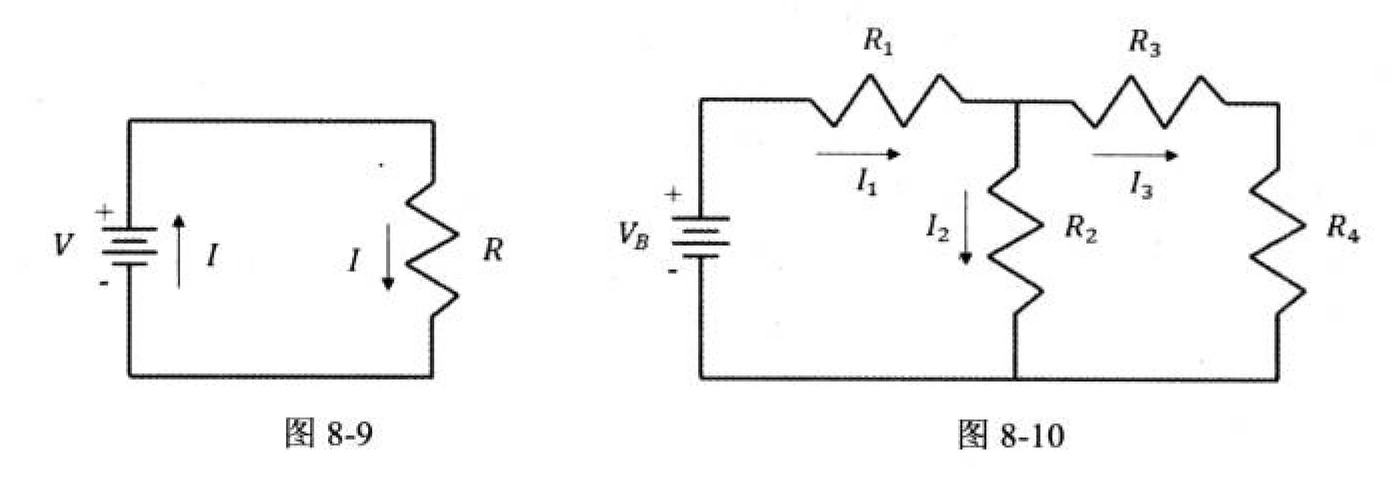


图 8-9　　　　　　　　图 8-10

我们任意选择电路网络中的一个回路，从回路上某一点出发，经过回路后，回到出发点。回路上每个元件上的电压之和必须等于 0。这个规律从下面的常规现象不难得出：电阻上的电压是沿着电流下降的，电池上的电压是沿着电流上升的。这就是基尔霍夫的电压定律。电阻上的电压可以由欧姆定律求得。把基尔霍夫定律和欧姆定律应用于如图 8-10 所示的左侧回路上，得到以下方程：

$$V_B - R_1I_1 - R_2I_2 = 0 \tag{8-14}$$

同样，我们把这两个定律应用于如图 8-10 所示的右侧的回路上，得到以下的方程：

$$-R_3I_3 - R_4I_4 + R_2I_2 = 0 \tag{8-15}$$

注意，根据电压的正负号约定，R_2 上的电压为正，因为我们采用的回路方向是顺时针方向，即沿着电流 I_2 的方向。

现在，重新排列方程(8-13)~(8-15)，得到以下的方程组：

$$I_1 \quad -I_2 \quad -I_3 = 0 \tag{8-16}$$

$$R_1I_1 \quad +R_2I_2 \quad = V_B \tag{8-17}$$

$$R_2I_2 - (R_3 + R_4)I_3 = 0 \tag{8-18}$$

把这个方程组写成矩阵形式，得到以下公式：

$$\begin{bmatrix} 1 & -1 & -1 \\ R_1 & R_2 & 0 \\ 0 & R_2 & -(R_3+R_4) \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} 0 \\ V_B \\ 0 \end{bmatrix} \tag{8-19}$$

现在我们把方程(8-20)表示成一般的形式：

$$\mathbf{AX} = \mathbf{C} \tag{8-20}$$

式中 **A** 为系数矩阵，**X** 是自变量矩阵，C 是常系数矩阵。在这个系统中，未知量是电流 I_1，I_2，I_3，它们组成矩阵 **X**。我们利用矩阵运算，求出这个 **X** 矩阵。首先，在方程的两端乘上一个 $\mathbf{A}^{-1}$(记住，矩阵的运算顺序非常重要)，得到以下方程：

$$\mathbf{A}^{-1}\mathbf{AX} = \mathbf{A}^{-1}\mathbf{C} \tag{8-21}$$

回忆一下，$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$，它是一个单位矩阵，把它代入方程(8-21)，得到：

$$\mathbf{IX} = \mathbf{A}^{-1}\mathbf{C} \tag{8-22}$$

最后，我们还知道 **IX**＝**X**，把它代入方程(8-23)，得到：

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{C} \tag{8-23}$$

因此，这个问题的求解步骤为：

- 把线性方程组排列成为(8-20)的形式。
- 计算系数矩阵 **A** 的逆矩阵。
- 然后求逆矩阵与常量矩阵 **C** 的乘积。

现在我们就用 Excel 的 Solve 工具求解这个方程组，即求出每个回路上的电流。

解：

打开 Excel，根据图 8-11，在单元格里输入 3 个电阻的符号及相应的值。

接着在 E3:G5 单元格里输入系数矩阵 **A**，把矩阵中的电阻用相应的单元格表示，如图 8-12 所示，这样处理的一个好处是，我们可以很容易地修改输入表中的电阻值。

正确设置单元格的格式，结果如图 8-13 所示。

在 B7:B9 单元格里输入常量矩阵，并输入它的符号，如图 8-14 所示。

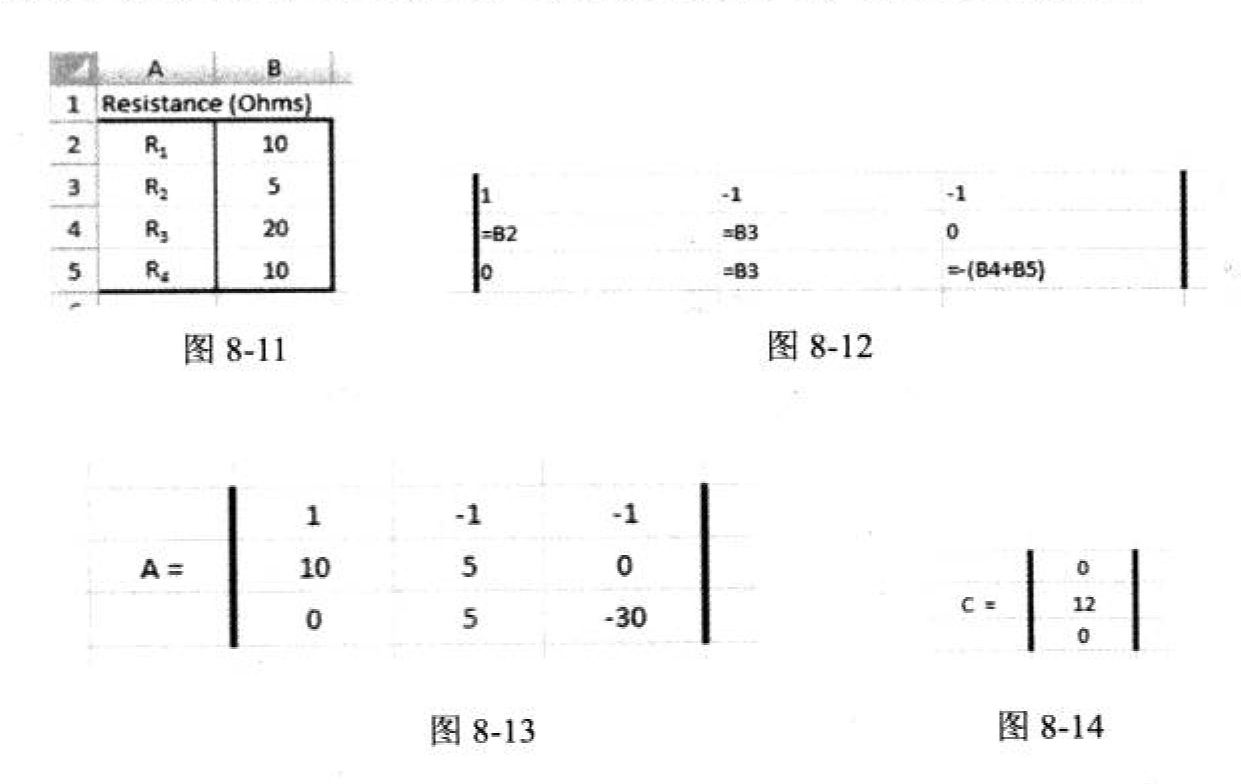


图 8-11　　图 8-12

图 8-13　　图 8-14

接着计算位于 E7:G9 区域的系数矩阵 **A** 的逆矩阵，然后输入符号，并设置它的格式，如图 8-15 所示。

现在，我们要计算逆矩阵与常量矩阵的乘积，把结果保存在 B11:B13 单元格里，如图 8-16 所示。

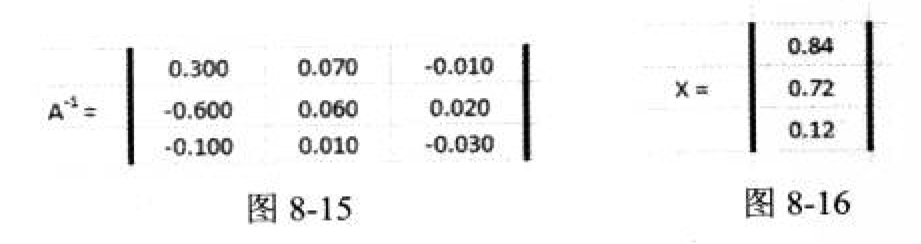


图 8-15　　图 8-16

最后，为了清楚说明程序的结果，我们加上一个输出表。在 D12:D14 单元格里输 I_1、I_2、I_3 电流符号，并把 **X** 矩阵里的值映射到 E12:E14 单元里，给表格加上一个标题，并设置表格的格式，最后得到的表格如图 8-17 所示。

现在来分析一下，当我们增加电路中的电阻 R_2 的值时，会有什么变化。我们故意把它

的值设为很大，把 R_2 的值改为 1kΩ。I_2 的值会是多少呢？I_2 的值减小到 0.009A。这是因为电流将沿着电阻较小的支路流动，因此，大部分电流没能经过 R_2，而流向电路的右侧回路里。

	A	B	C	D	E	F	G
1	Resistance (Ohms)						
2	R_1	10					
3	R_2	5			1	-1	-1
4	R_3	20		A =	10	5	0
5	R_4	10			0	5	-30
6							
7		0			0.300	0.070	-0.010
8	C =	12		A^{-1} =	-0.600	0.060	0.020
9		0			-0.100	0.010	-0.030
10							
11		0.84		Current (Amps)			
12	X =	0.72		I_1	0.84		
13		0.12		I_2	0.72		
14				I_3	0.12		
15							

图 8-17

8.3　教程：用 MATLAB 求解线性方程组

MATLAB 是一个功能强大的线性方程组求解工具，它特别适合求解大规模的线性方程组。与前面的 Excel 例子一样，我们用 MATLAB 的矩阵运算求解上述方程组。由于 MATLAB 是专为矩阵运算而设计的，它的运算效率较高，而且可以使用循环结构，因此用 MATLAB 可以求解需要千万次运算量的复杂方程组。下面介绍的求解方法对于小规模方程组也是非常有效的。对于大规模的线性方程组，还有效率更高的专用求解方法，如高斯消元法，这个问题超出了本书的范围。

在读者做下面的两个练习时，必须牢记这样一个思想：程序稍加修改就可以求解另一个问题，因此程序方法是最有用的。在这个问题是常量参数，在另一个问题也许是变量。因此在程序的开头定义参数，然后在程序中自始至终使用参数符号，这样，在将来修改时会非常方便。这种办法保证了未来使用时的灵活性。

在下面的两个练习里，我们求解电路中的电流。在例 8.4 里，我们把其中一个电阻当作变量，用循环计算，当电阻在某个范围变化时，电流的变化。最后画出这个电流变化的曲线图。如果读者还没有完成例 8.3 的练习，则必须仔细阅读它的说明，否则很难理解本例子中方程组是如何得到的。

例 8.4

用 MATLAB 计算如图 8-18 所示的每个分支的电流。

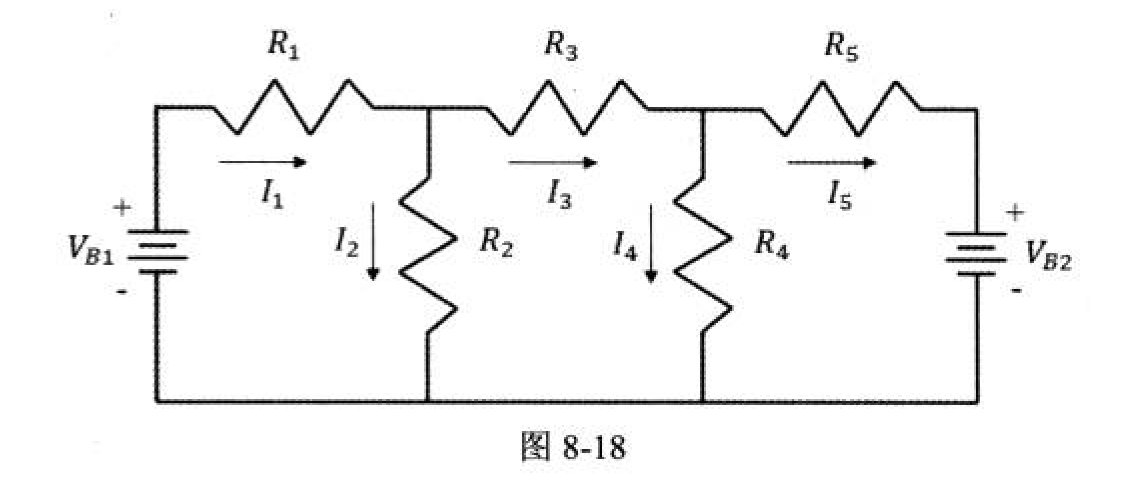


图 8-18

解：

应用基尔霍夫定律和欧姆定律，我们得到以下方程：

$$I_1 = I_2 + I_3 \tag{8-24}$$

$$I_3 = I_4 + I_5 \tag{8-25}$$

$$V_{B1} - R_1 I_1 - R_2 I_2 = 0 \tag{8-26}$$

$$R_4 I_4 - R_5 I_5 - V_{B2} = 0 \tag{8-27}$$

上述方程经化简后，得到如下的矩阵形式：

$$\begin{bmatrix} 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 \\ R_1 & R_2 & 0 & 0 & 0 \\ 0 & R_2 & -R_3 & -R_4 & 0 \\ 0 & 0 & 0 & R_4 & -R_5 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ V_{B1} \\ 0 \\ V_{B2} \end{bmatrix} \tag{8-28}$$

现在为求解这个方程组，我们在 MATLAB 里建立一个脚本文件。把电阻值定义为输入变量。新建一个 m-file 文件，输入以下的代码：

```
1   % Define the values of the resistors R1 - R5 in Ohms;
2   R1=5;
3   R2=25;
4   R3=12;
5   R4=6;
6   R5=15;
7
8   %Define the value of the voltage sources, V1 V2 in Volts;
9   V1=110;
10  V2=45;
11
12  %Define the coefficient matrix A, Row by Row;
13  A1 = [1 -1 -1 0 0];
14  A2 = [0 0 1 -1 -1];
15  A3 = [R1 R2 0 0 0];
```

```
16 A4 = [0 R2 -R3 -R4 0];
17 A5 = [0 0 0 R4 -R5];
18 A = [A1;A2;A3;A4;A5]
19
20 %Define the constants matrix C;
21
22 C = [0; 0; V1; 0; V2]
23
24 %Calculate the currents (X matrix);
25
26 X = inv(A)*C
```

把这个脚本文件保存为 resistor，并执行这个程序。我们发现 I_5 的电流值为-1.042A，这是计算出错，这表示在图 8-18 中假定的电流方向不符合实际情况。即 I_5 的实际流向与图 8-18 中的方向相反。

例 8.5

现在，我们改动这个电路，把 R_3 改为一个可变电阻。可变电阻在电路中经常用来控制电流大小，如控制收音机的音量，或者调节电路的其他特性。当 R_3 的电阻从 0.1Ω 变到 100Ω 时，用 MATLAB 计算图 8-18 中的电阻 R_3 上的电流变化。

解：

我们用一个 for 循环计算当可变电阻 R_3 从 0.1Ω 变化到 100Ω 时，流过 R_3 的电流值。并且画出电流随 R_3 变化的曲线。

对例 8.4 的脚本程序进行修改：

```
1    %Define the values of the resistors R1, R2, R4, and R5.in Ohms;
2    R1=5;
3    R2=25;
4    R4=6;
5    R5=15;
6
7    %Define the value of the voltage sources, V1 V2 in Volts;
8    V1=110;
9    V2=45;
10
11   %Vary the resistance of R3 from 0.1-100 ohms and find I
12   for i = 1:1000;
13         R3(i)=i/10;
14
15   %Define the coefficient matrix A, Row by Row;
16       A1 = [1 -1 -1 0 0];
17       A2 = [0 0 1 -1 -1];
18       A3 = [R1 R2 0 0 0];
19       A4 = [0 R2 -R3(i) -R4 0];
20       A5 = [0 0 0 R4 -R5];
21       A = [A1;A2;A3;A4;A5];
22
23   %Define the constants matrix C;
24       C = [0; 0; V1; 0; V2];
```

```
25
26 %Calculate the currents (X matrix);
27       X = inv(A)*C;
28
29 %Extract the currents from the X matrix;
30       I1(i)=X(1);
31       I2(i)=X(2);
32       I3(i)=X(3);
33       I4(i)=X(4);
34       I5(i)=X(5);
35 end;
36
37 %Make plot of the currents as functions of R3;
38 plot(R3,I1,R3,I2,R3,I3,R3,I4,R3,I5)
```

把这个脚本程序保存为 resistor_var，并执行这个程序。需要指出的是，当 R_3 的值到了 30Ω之后，系统对 R_3 变化的响应就不十分明显。从工程角度来看，这一点非常重要，一个电阻变化范围为 0.1~30Ω 的可变电阻比起一个 70Ω~100Ω 的可变电阻控制范围更大。

8.4　教程：用 Excel 求解非线性方程组

虽然有很多的工程问题需要求解线性方程组，但是也有许多工程问题需要求解非线性方程组。非线性方程不能用前面介绍过的方法来求解。然而，Excel 有一个重要的工具，规划求解方法，可以用来求解非线性方程组。

例 8.6

用 Excel 的规划求解方法求解下面的方程组：

$$3x_1^3 + 4x_2 = 109 \tag{8-29}$$

$$2x_1 + 7x_2 = 65 \tag{8-30}$$

解：

首先，我们改变一下方程的形式，把方程的全部项移动到等号的左边、0 移动到等号的右边(类似于第 6 章的求根方法)，并把每个方程赋给一个哑变量 y_1 和 y_2：

$$y_1 = 3x_1^3 + 4x_2 - 09 = 0 \tag{8-31}$$

$$y_2 = 2x_1 + 7x_2 - 65 = 0 \tag{8-32}$$

读者可能还记得，这样，方程的解就是求 x_1 和 x_2 的值使得 y_1 和 y_2 都为 0。

打开一个 Excel 电子表格，在单元格 A1:A2 里输入 x_1 和 x_2，在单元格 A4 和 A5 输入 y_1 和 y_2 符号。把 x_1 和 x_2 的初始估算值分别设置为 2 和 3。在 B4 和 B5 单元格里分别输入 y_1 和 y_2 的计算公式，与图 8-19 比较。

上述 x_1 和 x_2 的值是估算值。为了满足(8-31)和(8-32)的两个方程，y_1 和 y_2 的值必须变为 0。

当 y_1 和 y_2 的和为 0 时，方程也可能会得到满足。但是存在这样的情形：y_1 和 y_2 的和为 0，但是它们并不为 0(如-2+2=0)，如果$(y_1)^2$与$(y_2)^2$的和等于 0，则方程肯定也会得到满足。之所以使用平方，这是为了保证它们的值都为非负数，这样，当它们的和为 0 时，它们每个值肯定为 0。

接着求 y_1 和 y_2 的平方值，并求它们的和，对比图 8-20。

	A	B
1	x1	2
2	x2	3
3		
4	y1	-73
5	y2	-30

图 8-19

	A	B	C	D
1	x1	2		
2	x2	3		
3				y squared
4	y1	-73		5329
5	y2	-30		900
6				
7			SUM	6229

图 8-20

现在从 Ribbon 的数据选项卡里的数据分析组中选择规划求解工具，如图 8-21 所示。

如果找不到规划求解工具，可以用下面的办法：单击屏幕左上方的 Office 按钮，选择位于菜单底部的 Excel 选项，如图 8-22 所示。

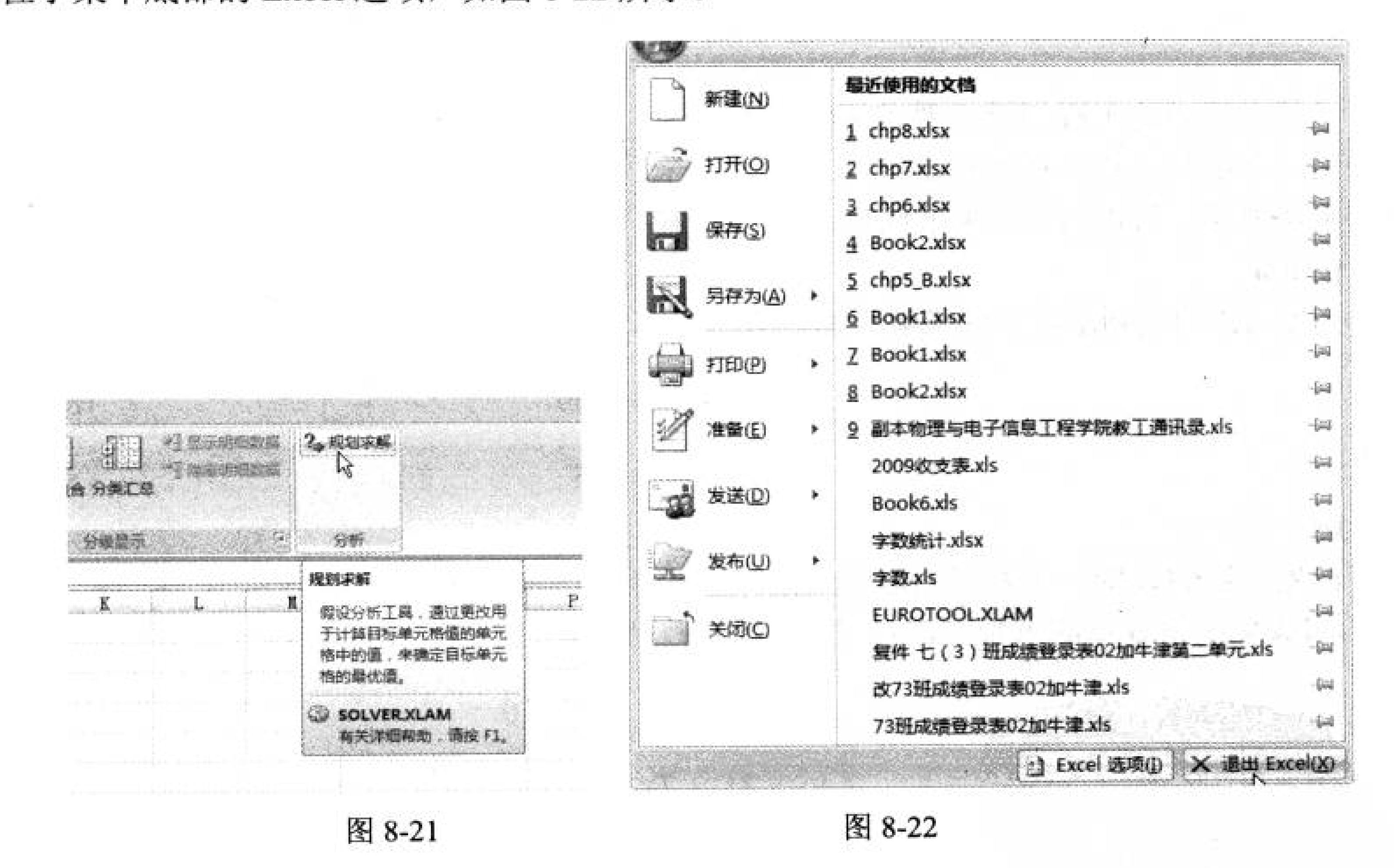


图 8-21　　　　图 8-22

接着，在弹出的【Excel 选项】对话框中选择【加载项】选项卡，如图 8-23 所示。

接着，从位于窗口底部的【管理】下拉列表中选择【加载】选项，然后单击【转到(go)】按钮，如图 8-24 所示。



图 8-23　　　　　　　　　　　　　　图 8-24

接下来会弹出一个【加载宏】对话框，选择【规划求解加载项】选项，并单击【确定】按钮，如图 8-25 所示。可能需要几分钟的时间完成规划求解工具的安装。

规划求解工具利用优化算法，通过改变其他单元格的值，让目标单元的值逼近于用户设置的目标值。这些可改变的单元称为可调节单元(规划求解工具最多允许 200 个可变单元)。在本例中，我们希望通过改变 x_1 和 x_2 的值(可变单元)，把包含$(y_1)^2$与$(y_2)^2$之和的单元逼近于 0。

选择规划求解工具后，出现规划求解参数对话框，把目标单元设置为 D7，把【等于】的可选项设置【值为】，并输入 0。把【可变单元格】设置为 B1:B2，然后单击【求解】按钮。如图 8-26 所示。

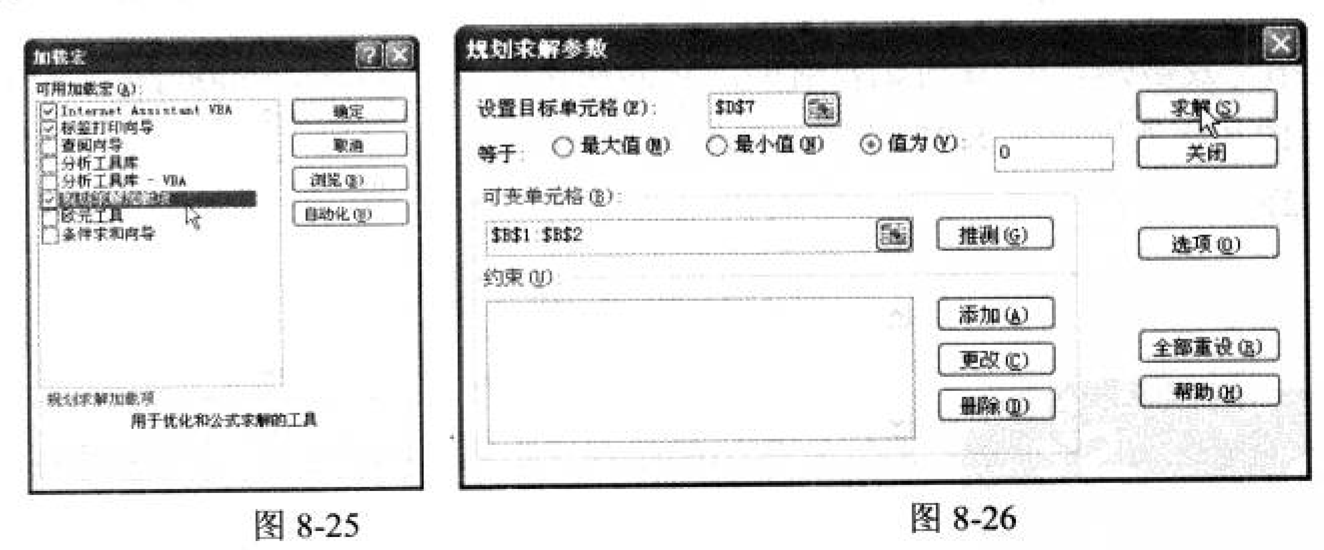


图 8-25　　　　　　　　　　　　　　图 8-26

规划求解工具得出一个可能的结果。精确的解是 $x_1=3$，$x_2=7$。单击【确定】按钮，接受它的结果。

记住，规划求解工具可能会得到多个解。它根据初始值进行收敛。要找到其他解，我们必须重新设置初始估算值(尽管本例只有一个解)。通常解的估算值可以通过图形方法求

得，或者如果对解的上限和下限有初步了解，可以利用插入法得到。用规划求解工具，包括约束条件，解决工程问题的更多例子将在本书的第10章里介绍。

8.5 习题

1. 用Excel的作图法，求下述方程的解：

(a) $3x+2y=11$
$5x-4y=33$

(b) $2x-5y=0$
$3x+4y=2$

2. 用Excel的单变量求解方法重做习题1。

3. 用Excel的作图法，求下述方程的解：

$$sinx+2y=4$$

$$3x^2+3y=6$$

4. 用Excel的规划求解方法，重做习题3。

5. 下面的方程表示两个圆的相交。用规划求解方法求这一组非线性方程的两对解。

$$(x_1-2)^2+(x_2-2)^2=16$$

$$(x_1-5)^2+(x_2-6)^2=2$$

提示：在纸上画出这两个圆，初步估算根的初始值。

6. 一发测试炮弹的初速度为1467ft/s(1000mph)，发射角为45°。第二发炮弹的发射地点离第一发炮弹的发射地点的距离为26400ft(5mile)，两者在同一个水平面上发射。第二发炮弹是在第一发炮弹之后Δt时刻发射的。它的初速度为2200ft/s(1500mph)，发射角度为60°。两者的几何位置如图8-27所示。两发炮弹的飞行轨迹都遵循本书第1章的方程(1-1和方程(1-2)：

$$h(t)=vt\sin\theta-\frac{1}{2}gt^2$$

$$x(t)=vt\cos\theta$$

式中，$h(t)$是飞行的高度，单位为ft，$x(t)$是水平距离，单位是ft。v是初速度，单位为ft/s，θ是发射角度。g是重力加速度(32.2ft/s^2)，t是飞行时间，单位为s。

第二发炮弹的发射时间是由以下条件来确定的：它正好拦截第一发炮弹。用规划求解方法求出第二发炮弹的发射时间，即Δt的值(以从第一发炮弹发射的时刻为起始时间)和拦截点的高度和离第一发炮弹发射点的水平距离。在同一个图上画出这两发炮弹的轨迹曲线。

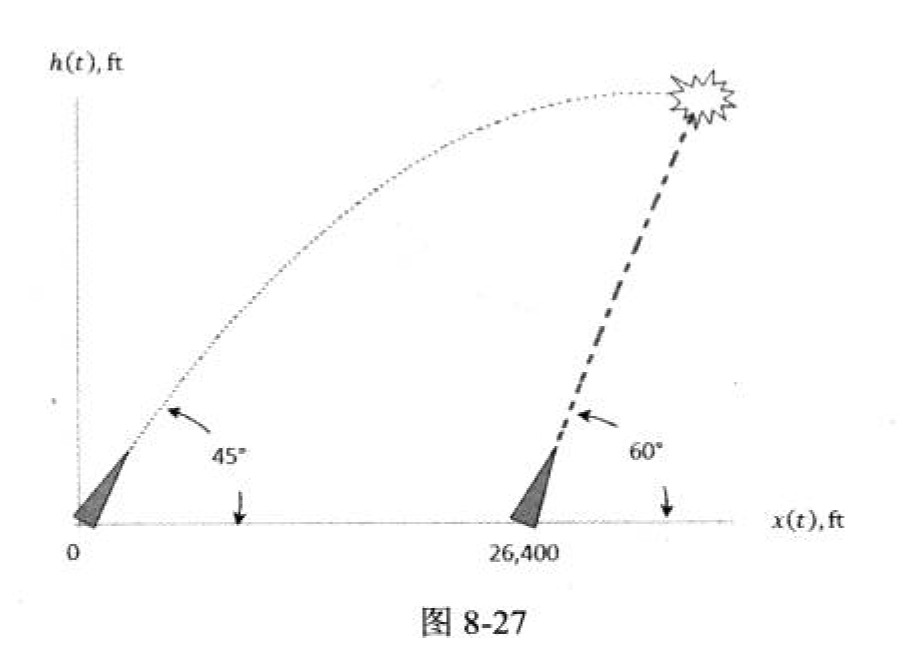


图 8-27

7. 一个由两个链杆组成的机器人如图 8-28 所示。

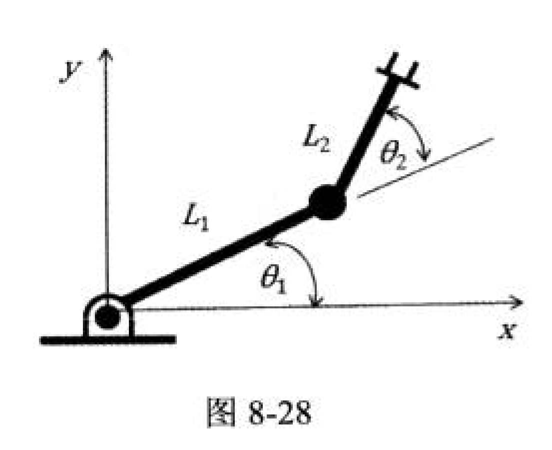


图 8-28

机器人手的位置是由下面的非线性方程组确定的：

$$x = L_1\cos(\theta_1) + L_2\cos(\theta_1 + \theta_2)$$

$$y = L_1 sin(\theta_1) + L_2\ \sin(\theta_1 + \theta_2)$$

如果机器人的第一个链杆的长度 L_1=18in，第二个链杆的长度 L_2=12in，用 Excel 解决以下问题：

(a) 要使 x_1=14in，y=12in，θ_1和θ_2角度必须为多大？按比例画出这个机器人图，验证自己的结果(用 CAD 软件)。注意，可能会有多个答案。

(b) 要使 x_1=－14in，y=12in，θ_1和θ_2角度必须为多大？按比例画出这个机器人图，验证自己的结果(用 CAD 软件)。注意，可能会有多个答案。

(c) 要使 x_1=29in，y=18in，θ_1和θ_2角度必须为多大？解释你的结果。

8. 在习题 7 的机器人上再增加第三个链杆，如图 8-29 所示。第三个链杆的长度为 6in。

(a) 写出描述 3 个链杆机器人手的位置的方程。

(b) 求出使手的位置到习题 7 的位置所需要的角度。同样用等比例图形验证自己的结果。

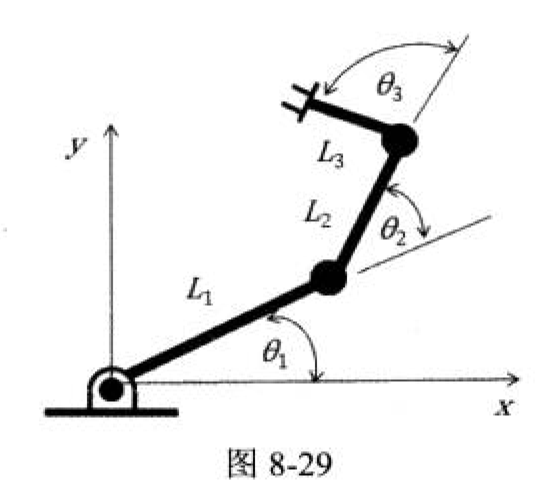


图 8-29

9. 用 Excel 求出以下线性方程组的解：

$$3x - 2y + 4z - 10 = 0$$

$$x + z = 2$$

$$x + 5y - 4z + 4 = 0$$

10. 用 Excel 求出以下线性方程组的解：

$$3x_1 + x_2 + 4x_3 - x_4 - 23 = 0$$

$$3x_1 + x_2 + x_3 - 2 = 0$$

$$-2x_2 + x_3 + x_4 = 0$$

$$x_1 + 5x_2 - 2x_3 + 4 = 0$$

11. 利用基尔霍夫定律列出图 8-30 的电阻电路的两条线性方程。证明，把基尔霍夫电压定律应用于左侧和右侧的环路得到的方程组与把基尔霍夫电压定律应用于左侧环路和右侧环路得到的方程组会得到 I_1，I_2，I_3 的同一组解。

(a) 根据下面的要求，写出 3 个方程，组成一个方程组。

- 在结点的电流定律。
- 左侧环路的电压定律。
- 右侧环路的电压定律。

(b) 根据下面的要求，写出 3 个方程，组成一个方程组。

- 在结点的电流定律。
- 左侧环路的电压定律。
- 外侧环路的电压定律。

(c) 用 Excel 求上述两组方程的解。

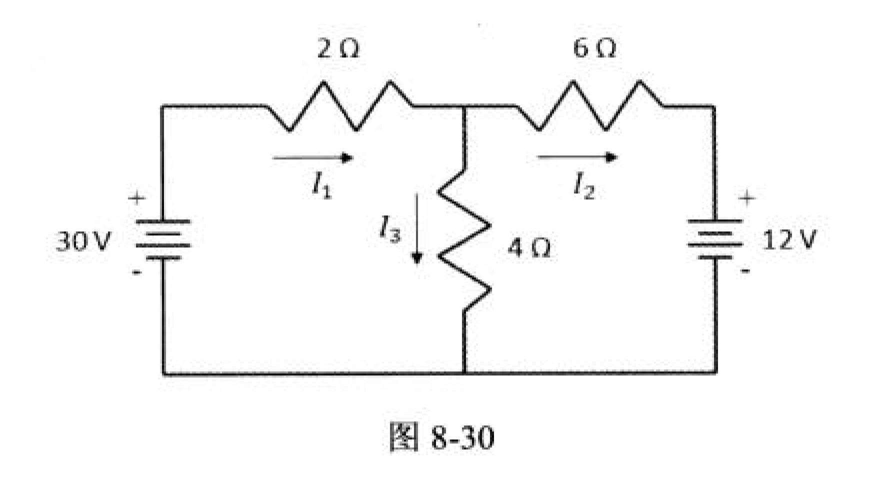


图 8-30

12. 用 MATLAB 计算图 8-31 的电阻电路中的 I_1~I_5 的值。图中的 R_1~R_6 分别取以下值：10Ω、2Ω、5Ω、10Ω、10Ω 和 20Ω。假设电池的电压为 30V。

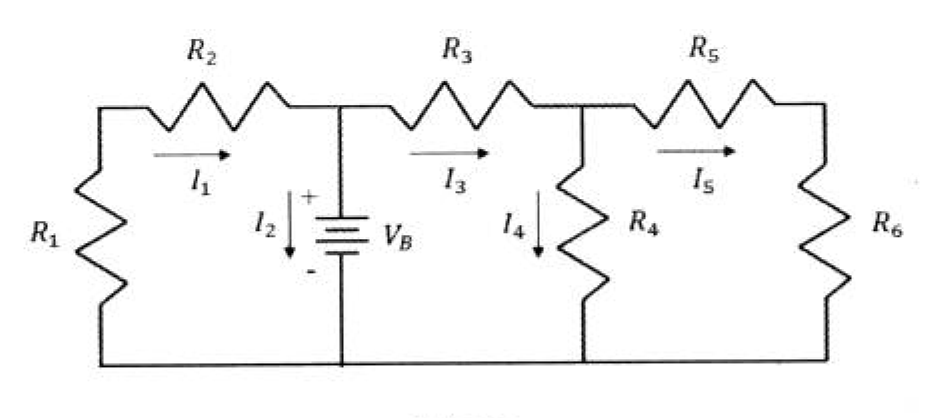


图 8-31

13. 在图 8-31 中，用一个可调电源代替 V_B。求出 I_1~I_5 与 V_B 的关系，并当 V_B 从 0V 变化到 10V 时，画出 I_1~I_5 的变化曲线。假设电池的电压可以连续调节，每隔 0.1V 计算一次，电阻的值与习题 12 相同。

14. 在图 8-31 中，用一个可调电源代替 V_B。求出 6 个电阻上的电压与 V_B 的关系，并当 V_B 从 0V 变化到 10V 时，画出它们的变化曲线。假设电池的电压可以连续调节，每隔 0.1V 计算一次，电阻的值与习题 12 相同。

15. 一个 10003 的物体由 3 根电缆支撑，如图 8-32 所示。注意，电缆 3 固定在 y 轴上，离原点 O 的距离为 d，利用矢量分析和物体平衡的原理(通常在力学中介绍)，列出 3 个方程。以 3 根电缆的拉力 T_1、T_2 和 T_3(单位为磅)为自变量，得到以下方程：

$$0.348T_1 - 0.557T_2 = 0$$

$$-0.348T_1 - 0.239T_2 + \left(\frac{d}{L}\right)T_2 = 0$$

$$0.870T_1 + 0.796T_2 + \left(\frac{10}{L}\right)T_3 - 1000 = 0$$

式中，$L=\sqrt{10^2 + d^2}$，是第三根电缆的长度，单位为 in。

编写一个 MATLAB 程序，当 d 为下列值时，求这 3 根电缆的拉力：

(a) 0 in

(b) 5in

(c) 10in

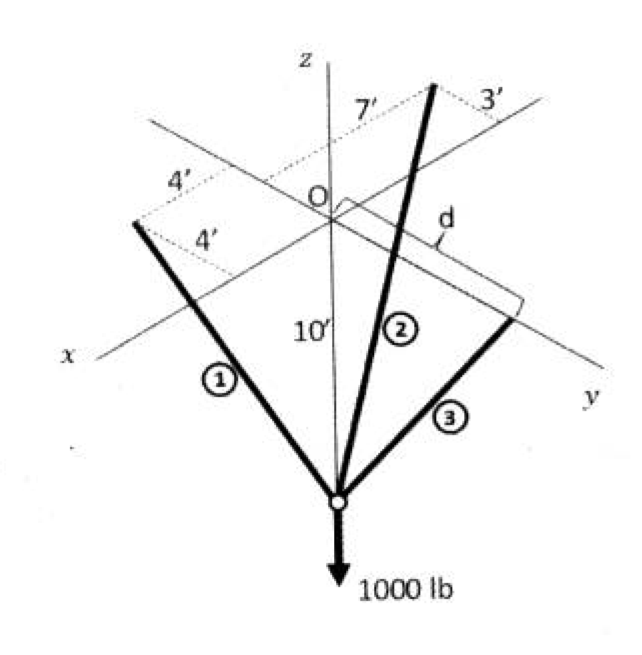


图 8-32

16. 假设习题 15 中，当物体的重量到达 1500bl 时，电缆就有可能断裂。假设把安全系数定义为 1500bl 除以 3 个电缆的最大拉力，编写一个 MATLAB 程序，*d* 从 0 变化到 15in，间隔为 0.1in，对于每个 *d* 的值，计算安全系数。画出安全系数与 d 的关系曲线图。当 *d* 值取多大时，安全系数最高。

第 9 章

数 值 积 分

引言

工程技术专业的学生在大学期间需要学习很多的数学课程，参加实际工作的工程师们每天都要用到数学知识。微积分是大多数工程理论的基础。本章将向读者介绍如何进行数值积分。即使读者没有学过微积分，也应该能理解本章几个教程的操作步骤，而且能掌握其中一种方法，它可以帮助读者更好地理解微积分并可以用来解决实际中的工程问题。

本章读者要学习以下内容：

- 掌握微分和积分的基本概念。
- 如何用 Excel 和 MATLAB 进行数值积分计算。
- 如何通过增加步长得到一个比较准确的解和收敛到一个准确解。
- 通过数值积分求解一个来自统计领域的问题。

9.1 微积分概念

微积分的两个核心内容是微分运算和积分运算。微分运算与求一个函数的变化率有关。例如，如果我们知道了汽车的速度是时间的函数，则可以求得汽车的加速度(速度对时间的变化率)。在另一方面，如果已知加速度(由于加速度与力成正比，在测试中经常使用一个不是很昂贵的、被称为加速器的仪器)。如果我们求出加速度对时间的积分，就得到速度的时间函数。为了求得速度，还要知道边界条件，如初始速度。我们把上述的关系用数学公式来表示：

$$a = \frac{dv}{dt} \tag{9-1}$$

和：

$$v = \int a dt \tag{9-2}$$

这两个公式分别表示“加速度是速度对时间的微分”和“速度是加速度对时间的积分”，在高等数学的课程里，读者将会学到如何用规则、表格数据和数值方法求微分和积分。例如，当读者需要求一个多项式某一项的微分时，就可利用这样一条规则：把它的次数减 1 再乘上原来的系数，就是它的微分。下面的公式正是利用这条规则：

$$y' = \frac{d}{dx}(x^3 - 3x^2 + 5) = 3x^2 - 6x \tag{9-3}$$

注意，这里的 y'读成“y 的导数”(y prime)，一个数的导数表示它对自变量的微分(在本例是时间)。当我们对一个多项式积分，应把每一项的指数加上 1，再除以新的指数值。例如：

$$\int (3x^2 - 6x)dx = x^3 - 3x^2 + C \tag{9-4}$$

注意，微分与积分是紧密相联系的。但是，求积分得到的表达式并不与求微分时的表达完全相同。原来微分的表达式有一个常量 5，而微分后公式(9-4)里有一个 C 常量。这是因为，在微分时，任何常量的微分都为 0。因此，有微分值相同的多项式会有无限个。我们求积分，就是求一个符合微分公式的函数。答案是不唯一的：只有知道边界条件或初始条件，才能得到完整的解。

通过分析函数的图形，我们可以进一步了解微分与积分的关系。在图 9-1 里，画出了 $y=x^3-3x^2+5$ 函数的曲线，横坐标的范围为-2~4。

微分就是函数对输入变量的变化率。从图形上看，y 对 x 的变化率就是函数的斜率。对于方程(9-3)而言，导数 y'的公式是 $3x^2-6x$。用这个公式可以求任意点的斜率。例如，对于 $x=1$，它的斜率是$(1)^3-6(1)=-3$，图 9-2 说明了该点的斜率。如果把公式中的常量 5 换成另一个值，对图形的影响不过是把曲线向上或向下平移。但是，曲线在任意点的斜率保持不变。因此不管常量值取多少，函数的微分总是不变的。

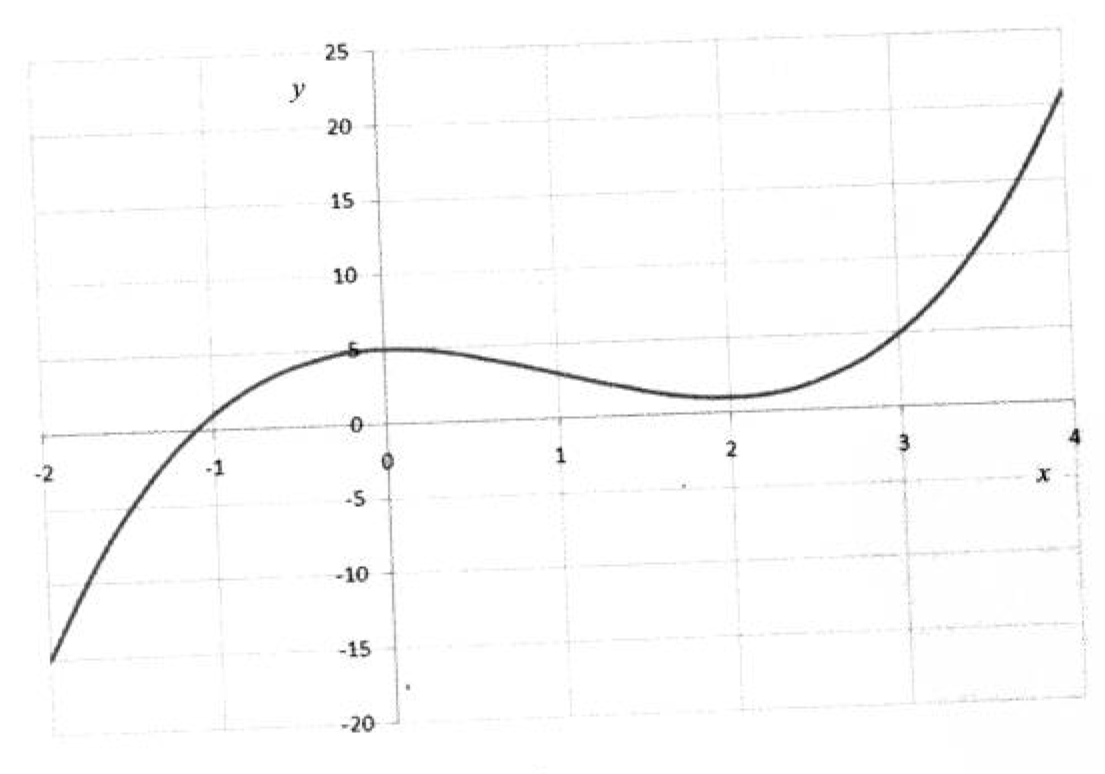


图 9-1

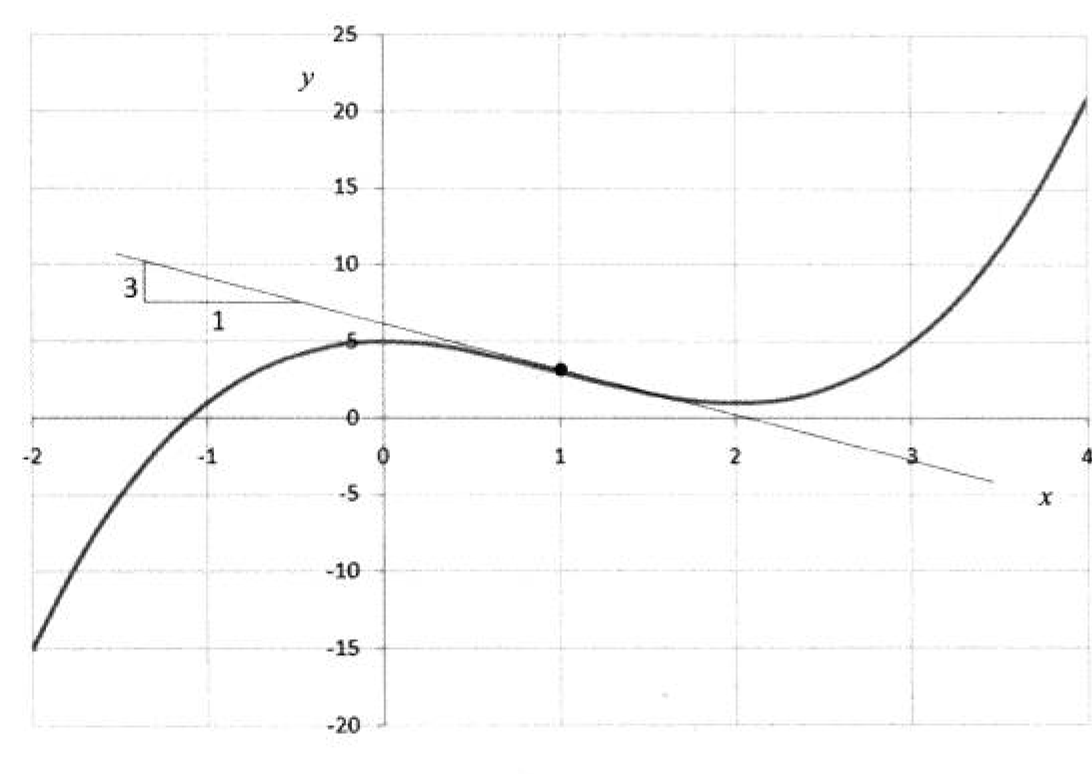


图 9-2

我们通常对哪一点的斜率为 0 这个问题很感兴趣。令微分 $3x^2$-6 等于 0，就得到当 x=0 或 x=2 时，函数的斜率为 0。这些位置非常重要，因为它们代表函数的极大值或极小值的位置。从图形上看，在 x=0 位置有一个局部极大值，在 x=2 位置有一个局部极小值。从数学上讲，要确定极值点是极大值还是极小值，需要再次求微分。如果二阶微分的值是正数，则该点是局部极小值，如果二阶微分值小于 0，则该点是局部最大值，反之则为局部最小值。在本例里，二阶导数 $y''=6x-6$。当 x=0 时，y''=-6,因此，该点是局部极大值，当 x=2 时，y''=6,因此该点是局部极小值。

虽然微分可以逐点求值，但是积分必须在 x 的一个范围内求值。像公式(9-4)那样的积分公式被称为不定积分，因为没有定义积分的范围。当积分的范围确定后，则积分公式就成为定积分。例如，考虑函数 $y=3x^2-6x$ 在 x=0~4 范围内的积分，我们把这个积分表示为如下的形式：

$$\int_0^4 (3x^2 - 6x)dx \tag{9-5}$$

从数学上讲，把上限值(4)代入公式(9-4)的不定积分公式减去公式(9-4)在下限的值就是定积分的值：

$$\int_0^4 (3x^2 - 6x)dx = (x^3 - 3x^2)\Big|_0^4 = (4^3 - 3(4)^2) - ((0)^3 - 3(0)^2) = 16 - 0 = 16 \tag{9-6}$$

当然，这个结果的单位取决于 x 和 y 的单位。我们注意到，方程(9-4)中的常量 C 已经被忽略了，因为，C 同时出现在上限和下限的积分值，两者相减，C 就被抵消了。

定积分的图形解释是函数的曲线在某个间隔内的面积。公式(9-6)的积分计算可以用图 9-3 来说明，在曲线之下，x 轴之上的面积为正面积，而在 x 轴之下曲线之上的面积为负面积。

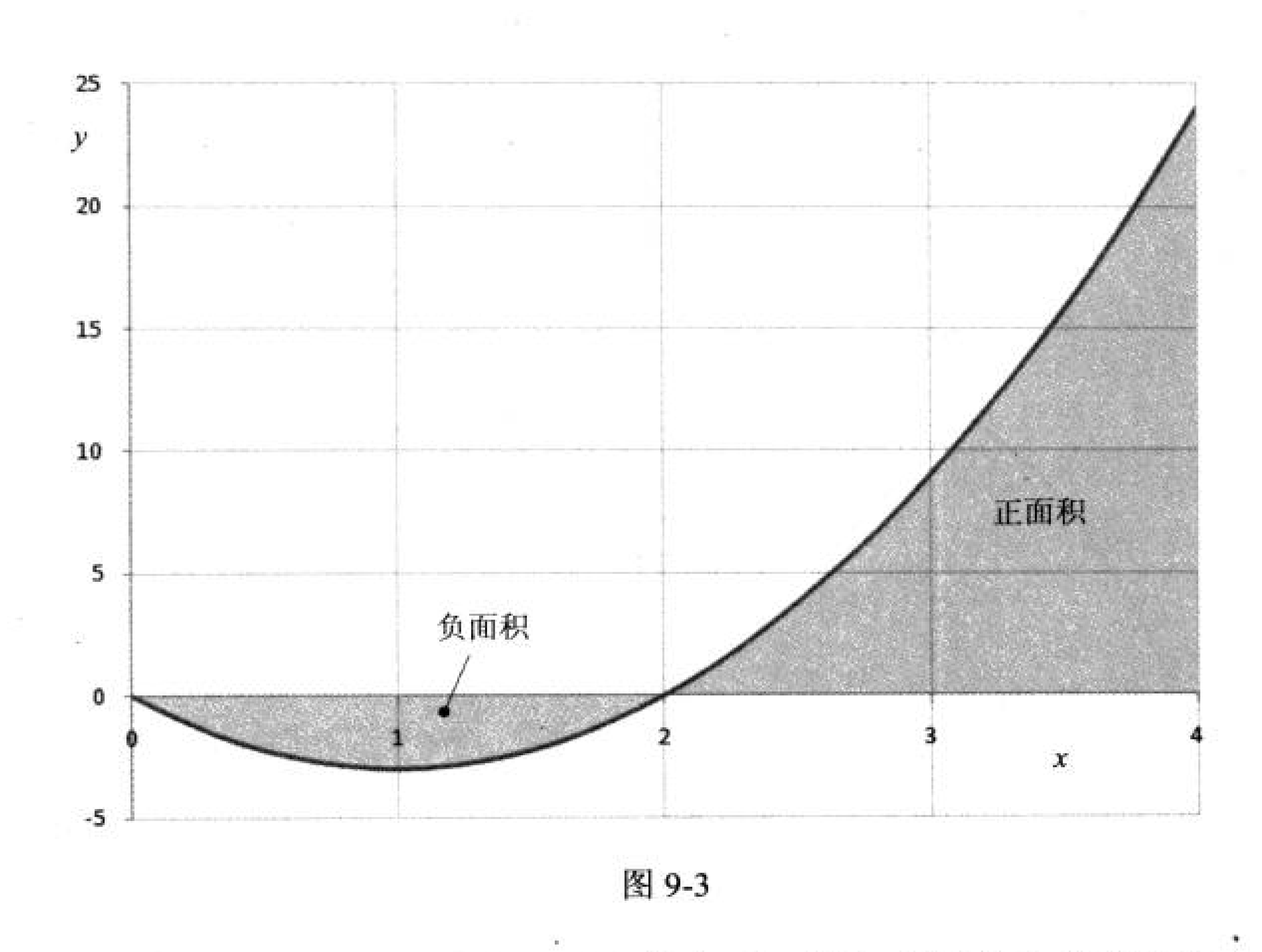


图 9-3

求多项式和其他许多简单函数的积分比较直观。微积分教材和其他很多参考书里提供了许多积分公式。但是有时，我们会遇到一些不存在已知的解析解的积分。换句话说，这些函数的积分不能通过积分公式得到，而只能通过其他措施。在这些情况下，我们要用下面即将介绍的数值法近似表示它们的积分值。

9.2 教程：函数的数值积分

例 9.1

在 Excel 里用梯形法求(9-4)积分公式的近似值。我们重复写出此积分公式：

$$\int_0^4 (3x^2 - 6x)dx$$

解：

根据梯形法，某个曲线的积分面积近似表示为许多个梯形面积之和。考虑图 9-4 所示的曲线，x_1 和 x_2 是自变量 x 的两个值，相应的 y 值为 y_1 和 y_2。用直线连接点(x_1,y_1)与点(x_2,y_2)，构成一个梯形，底边长为 x_2-x_1，平均高度为$(y_1+y_2)/2$，如图 9-5 所示。因此这个梯形的面积如图 9-5 所示。

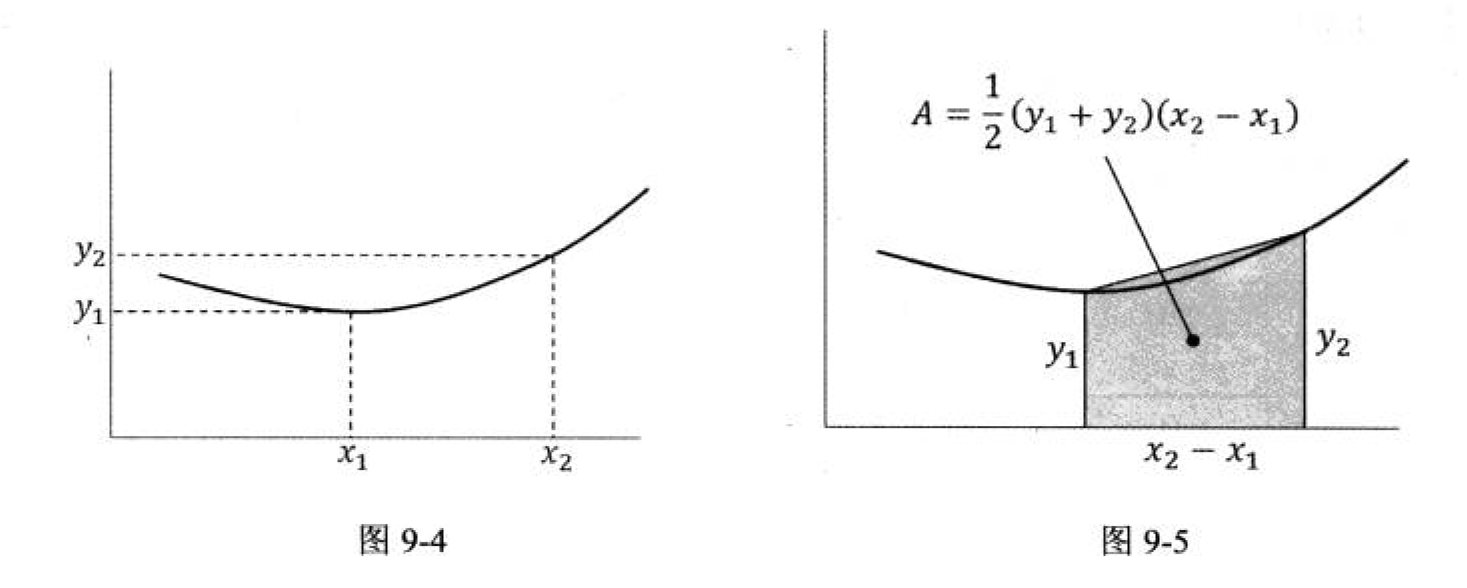


图 9-4　　　　图 9-5

当然，只有当这段曲线为直线时，用梯形法求得的面积才是它的准确值。但是用多段直线近似表示曲线，我们可以得到一个与准确面积相当接近的值，如图 9-6 所示。

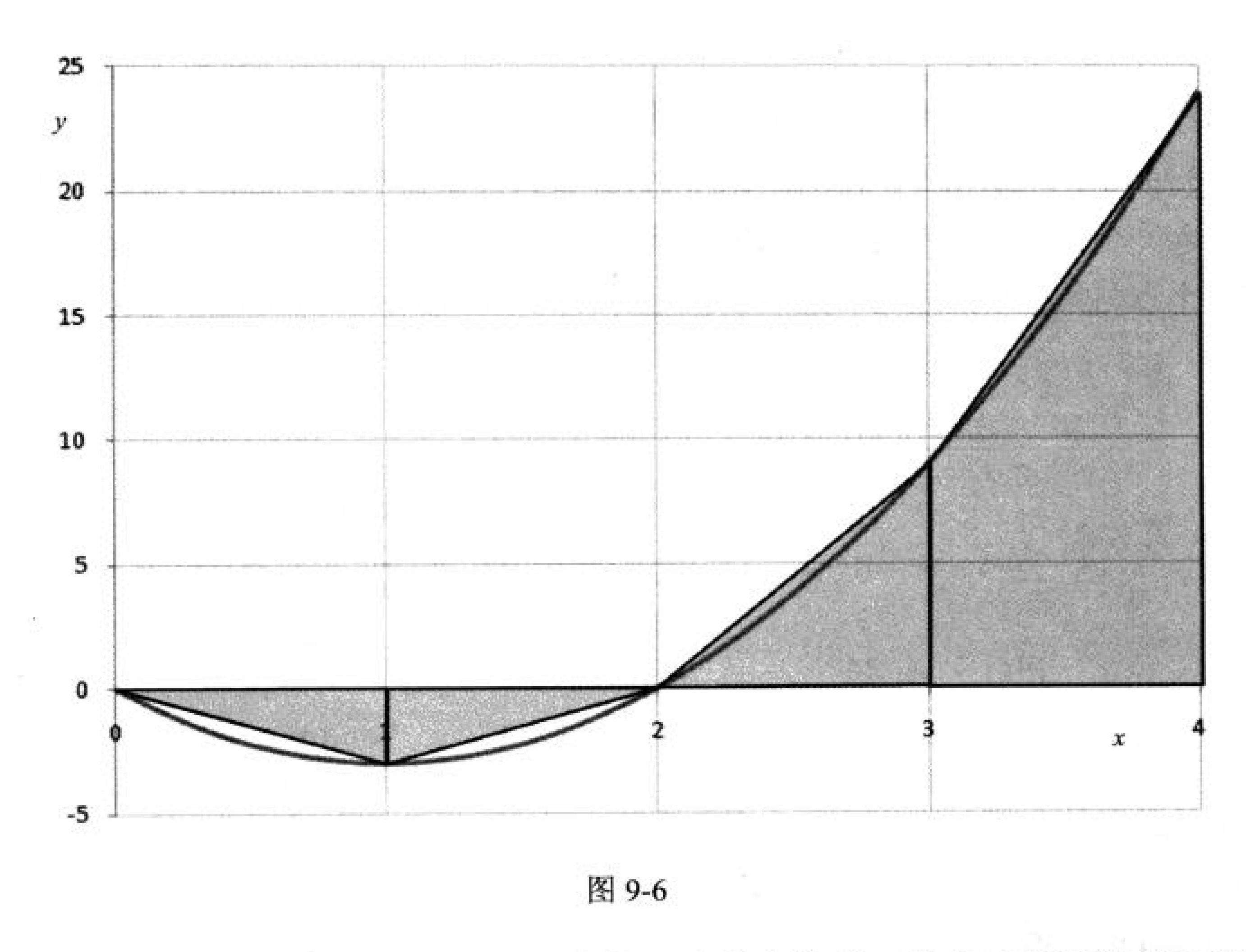


图 9-6

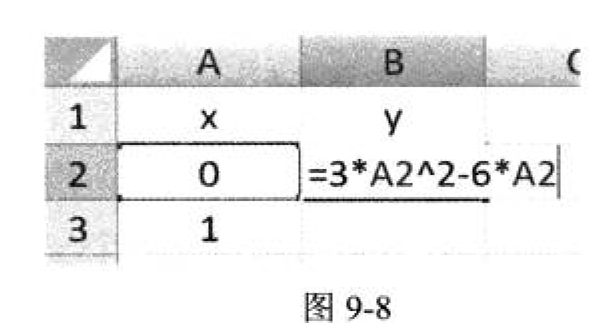
还有其他方法可用于数值法求积分。例如，辛普森法用二阶多项式近似表示曲线。虽然辛普森法用较少的间隔点就可以得到更准确的结果，但是梯形法简单、容易实现。由于计算机的运算速度越来越快，对于大多数工程问题来说，并不需要考虑运算的效率。用梯形法，并且在积分范围内取大量间隔点，既不会显著增加运算，又能得到相当精确的结果。

打开 Excel，新建一个工作表。在 A1 和 B1 单元里，分别输入 x 和 y 符号。在 A2 和 A6 单元里，输入 0、1、2、3 和 4，如图 9-7 所示。在第一个近似计算积分的例子里，我们取 x 的积分间隔为 1 个单位。

在单元 B2 里，输入 y 的公式，如图 9-8 所示。

	A	B
1	x	y
2	0	
3	1	
4	2	
5	3	
6	4	
7		

图 9-7

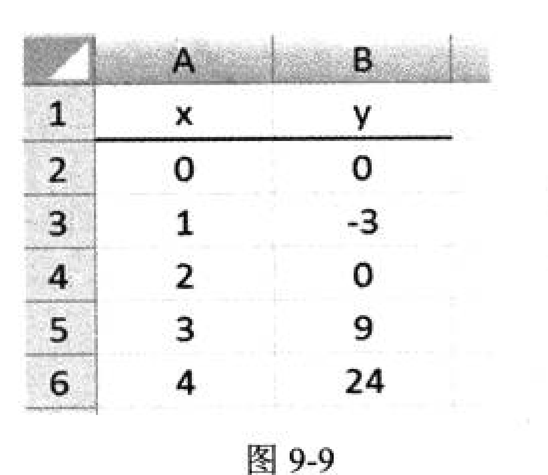


	A	B
1	x	y
2	0	=3*A2^2-6*A2
3	1	

图 9-8

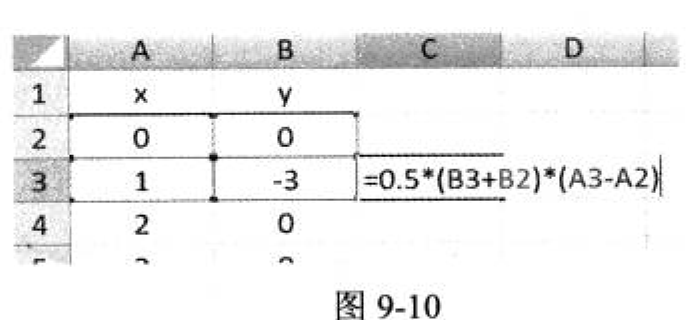
把 B2 单元里的公式复制到 B3:B6，如图 9-9 所示。

在 C3 单元里，输入第一个梯形面积的计算公式，如图 9-10。注意，我们还没有在 C2 单元里输入一个公式。本例有 4 个梯形区域，有 5 个点表示区域边界。因此，第一个区域是在第一个点与第二个点之间。



	A	B
1	x	y
2	0	0
3	1	-3
4	2	0
5	3	9
6	4	24

图 9-9

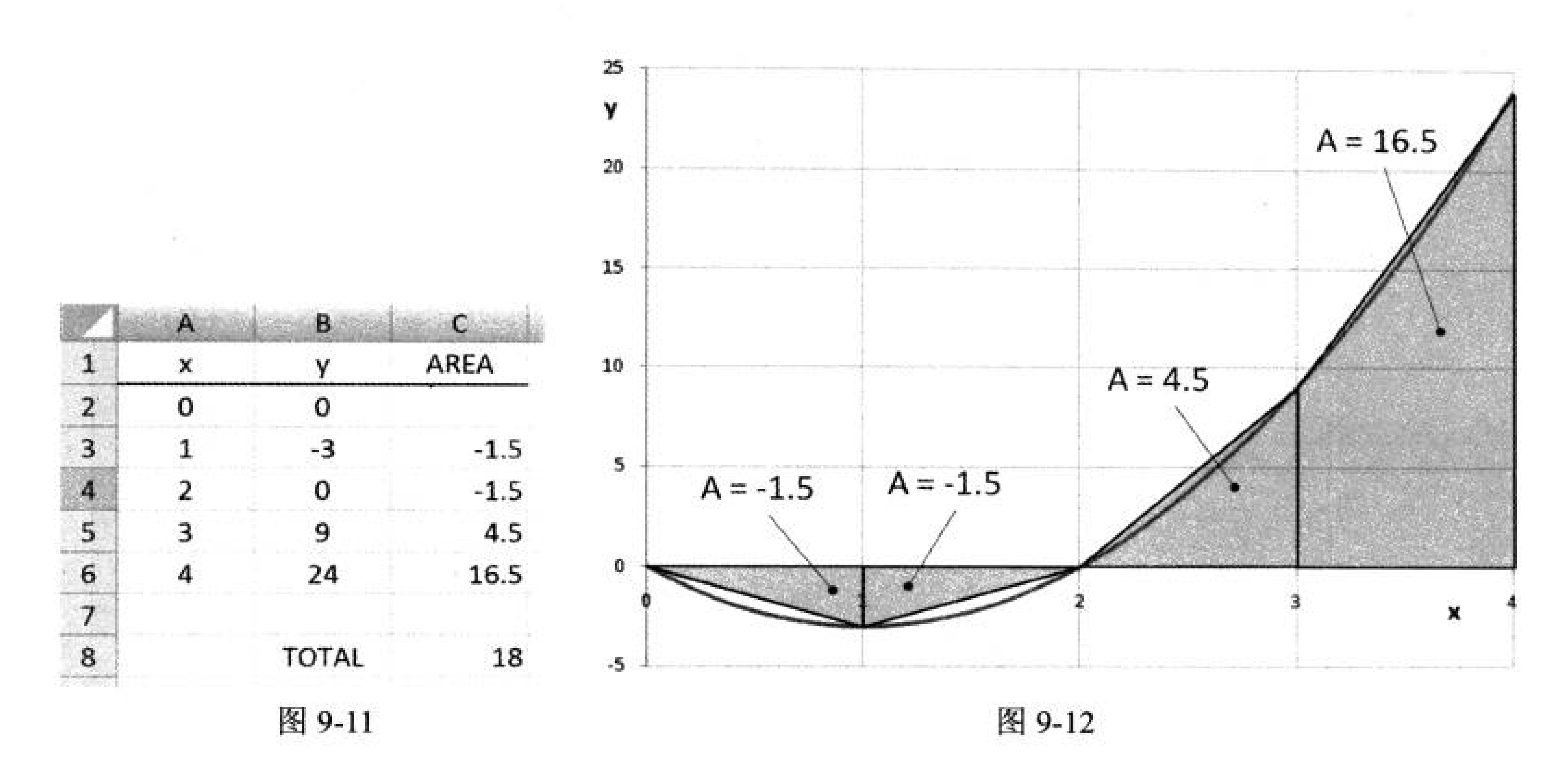


	A	B	C	D
1	x	y		
2	0	0		
3	1	-3	=0.5*(B3+B2)*(A3-A2)	
4	2	0		

图 9-10

把此单元里的公式复制到 C4:C6 单元里，在单元 C8 里，输入公式“=sum(c3:c6)”，求出全部梯形面积之和，如图 9-11 所示。

在这个近似计算过程中，我们把 x 的增量设置为 1，求得的积分值为 12(前面曾说明过，这个积分值的单位取决于 x 和 y 的单位)。在前一节里，我们用解析法，求得它的积分值为 16。相对误差达 12.5%。图 9-12 显示了这个近似过程使用的 4 个梯形。可以看出，在 x=2 到 x=4 之间积分面积为正值，梯形法得到的面积比实际面积大，而在 x=0 到 x=2 之间积分面积是负值，而梯形法得到的面积比实际值小。



	A	B	C
1	x	y	AREA
2	0	0	
3	1	-3	-1.5
4	2	0	-1.5
5	3	9	4.5
6	4	24	16.5
7			
8		TOTAL	18

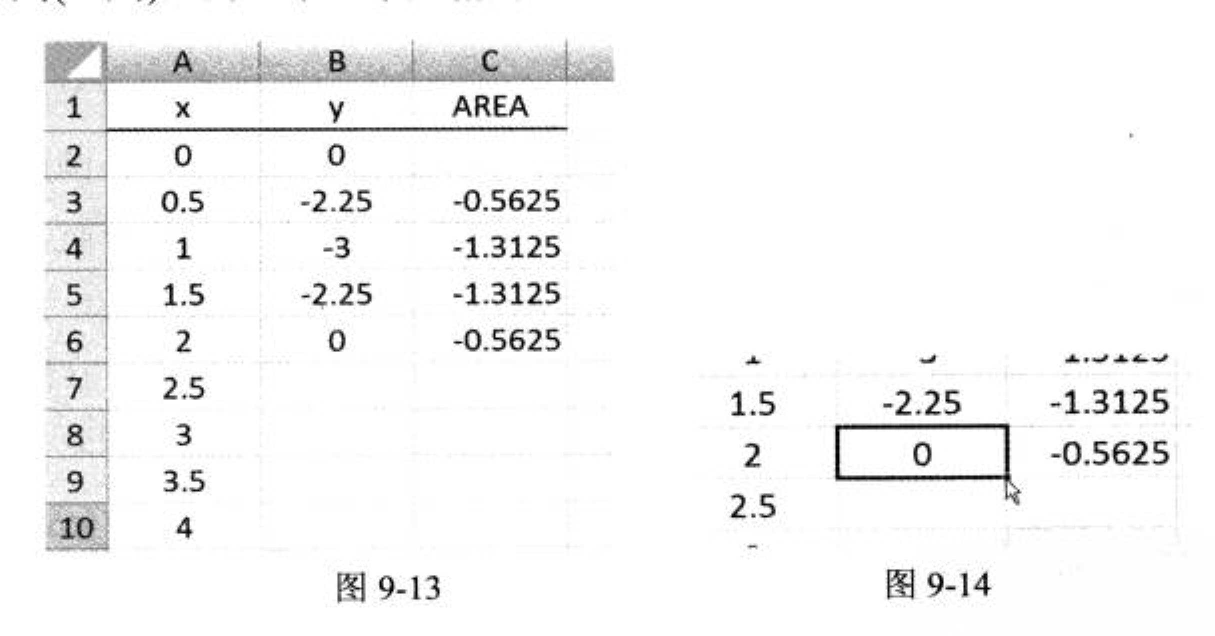
图 9-11　　　　图 9-12

我们增加梯形个数，以提高其运算精度。现在我们来修改前面的电子表格，把区间从 4 个增为 8 个。

删除 TOTAL 单元格的内容和面积之和。在 A 列里，改变 x 的值，取从 0~4 的增量为 0.5，如图 9-13 所示。

注意，B 列和 C 列的公式仍然有效，只需要把它们复制到下面的单元格里。

双击 B6 的填充柄，如图 9-14 所示。把 B6 单元里的公式复制到下面的单元格里，直到左边相邻列(A 列)里的一个空单元格为止。



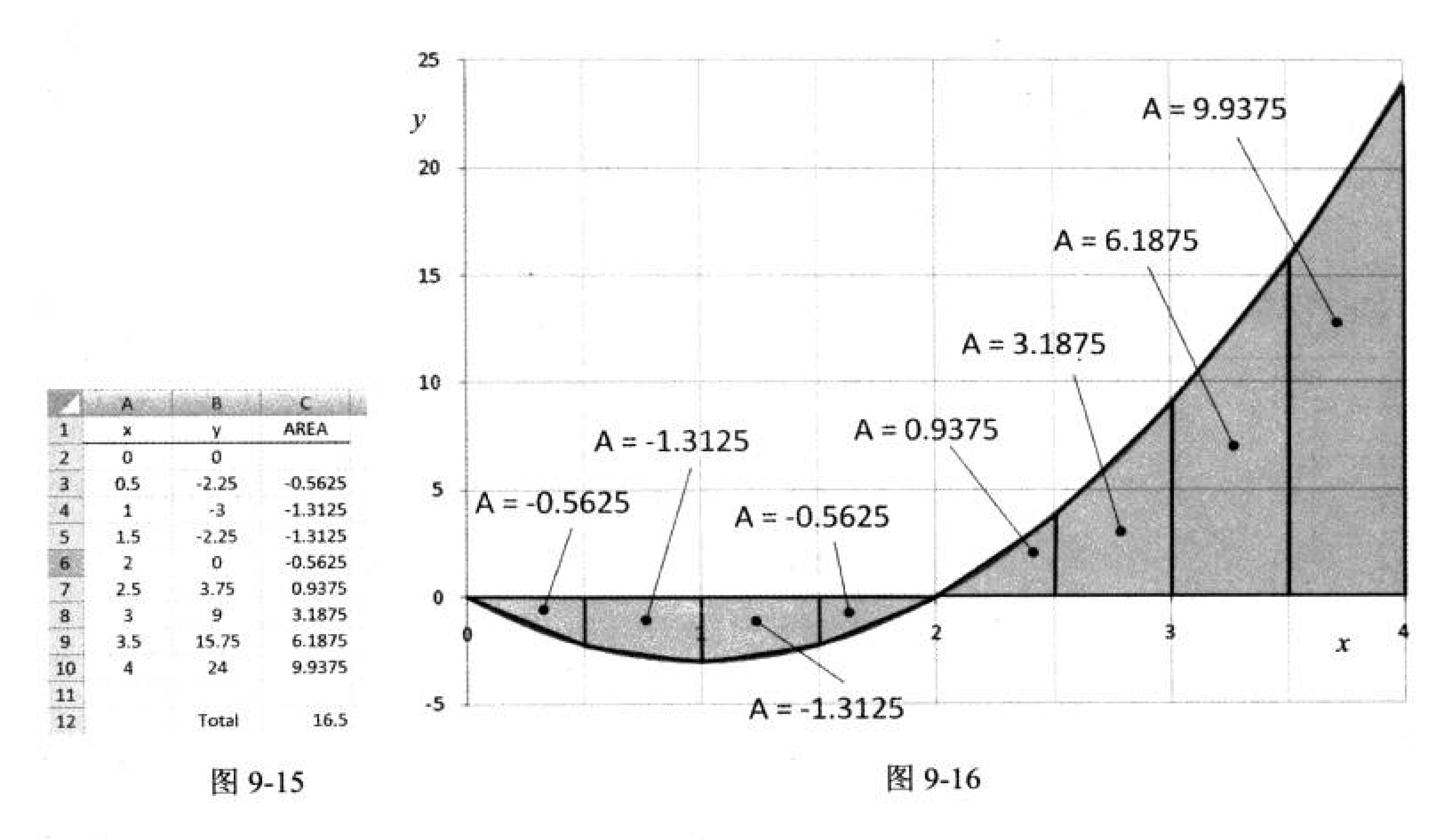
	A	B	C
1	x	y	AREA
2	0	0	
3	0.5	-2.25	-0.5625
4	1	-3	-1.3125
5	1.5	-2.25	-1.3125
6	2	0	-0.5625
7	2.5		
8	3		
9	3.5		
10	4		

1.5	-2.25	-1.3125
2	0	-0.5625
2.5		

图 9-13　　　　图 9-14

采用同样方法，把 C6 单元格里的公式复制到下面的单元格里。在 C 列里，增加一个单元，求 C 列的面积之和，如图 9-15 所示。

现在得到的结果是 16.5。把区间从 4 个增加到 8 个，相对误差从 12.5%减少到 3%左右。图 9-16 显示了该近似过程中的各个梯形面积。与图 9-12 相比较，显然可以看出，增加区间得到的结果接近准确的曲线面积。



	A	B	C
1	x	y	AREA
2	0	0	
3	0.5	-2.25	-0.5625
4	1	-3	-1.3125
5	1.5	-2.25	-1.3125
6	2	0	-0.5625
7	2.5	3.75	0.9375
8	3	9	3.1875
9	3.5	15.75	6.1875
10	4	24	9.9375
11			
12		Total	16.5

图 9-15　　　　　　　　图 9-16

例 9.2

在 MATLAB 里用梯形法求(9-4)积分公式的近似值。

我们重复写出此积分公式：

$$\int_0^4 (3x^2 - 6x)dx \tag{9-4}$$

解：

在 Excel 里很容易输入数值积分公式，但是存在一个问题：当需要更多的步长时，就必须对电子表格做大范围的修改。在这个例子里，我们知道准确的答案，因此能够确定 8 个区间就可以求得一个比较准确的结果。实际上，当我们用数值法求积分时，通常并不知道其结果，因此我们如何知道求得的结果是否准确？一种方法是增加步长，直到结果收敛到某个值为止。使用 MATLAB 的一个优点是，可以建立一个脚本文件，在脚本文件里可以方便地修改区间个数。在本例里，我们要建立一个 MATLAB 函数求积分值。这个函数的参数是间隔数。

图 9-17 是这个函数的流程图。我们首先输入间隔数 k。对于 k 个间隔，一共有 k+1 个点。第一个点的坐标为(x(1)，y(1))，它们需要在循环之前定义。在循环体内，我们首先要求 x(2)和 y(2)，并求由这两个点组成的梯形的面积。把这个面积累加到 *SUM* 变量上。*SUM* 代表积分值，在循环之前已初始化为 0。重复执行这个过程，直到最后一个间隔为止。

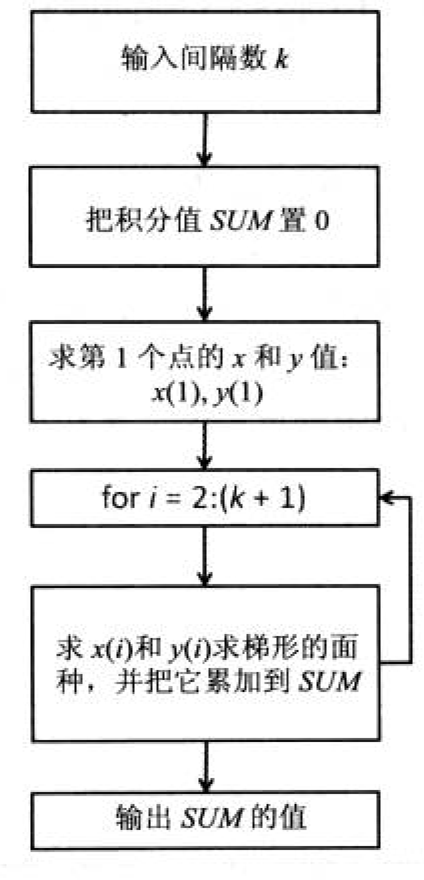


图 9-17

在 MATLAB 里建立一个函数，输入以下代码：

```
1  function SUM = intexample(k)
2  % Computes the integral of y=3x^2-6x for
   x  = 0 to 4
3  % k = Number of intervals
4
5  % Initialize the SUM (value of the integral)
6  SUM = 0;
7  %Calculate the increment value
8  increment = 4/k;
9  %Set the values for the first endpoint
10 x(1) = 0;
11 y(1) = 0;
12 % Calculate x and y values at the end of each interval
13 % Calculate the area for the interval, add to SUM
14 for i = 2:(k+1)
15     x(i) = x(i-1) + increment;
16     y(i) = 3*(x(i)^2)-6*x(i);
17     SUM = SUM + .5*(y(i) + y(i-1))*(x(i) -(i-1));
18 end
```

把这个 m-file 保存为 intexample，在命令提示符后，输入 4 个间隔数，计算这个函数的值。

```
> intexample(4)
ans =
      18
```

我们注意到结果为 18，与前面的 Excel 例子在间隔数为 4 的结果相同。在用更多的间隔数再来计算这个积分值之前，我们先仔细分析它的流程图和 MATLAB 程序。特别要注意变量 k 和 i。k 是一个输入变量，在本例里，它的值设置为 4。变量 i 表示区间点的索引。当 i=1 时，x 的值是积分的下限(0)，y 的值就是函数在 x=0 的值(在本例里，y(1)的值也正好为 0)。定义并保存了 x 和 y 的初始值后，程序开始进入 for 循环体。在每个区间中，这个循环都会被执行一次，因此循环的次数为 k 次。由于循环内需要计算的第一个点是 i=2 的位置，因此把这个循环的初始值设为 2，终值为 k+1，本例它的值为 5，每次循环，计算一个梯形的面积，并把结果累加到 *SUM* 变量里，该变量在循环之前已初始化为 0，如图 9-18 所示。

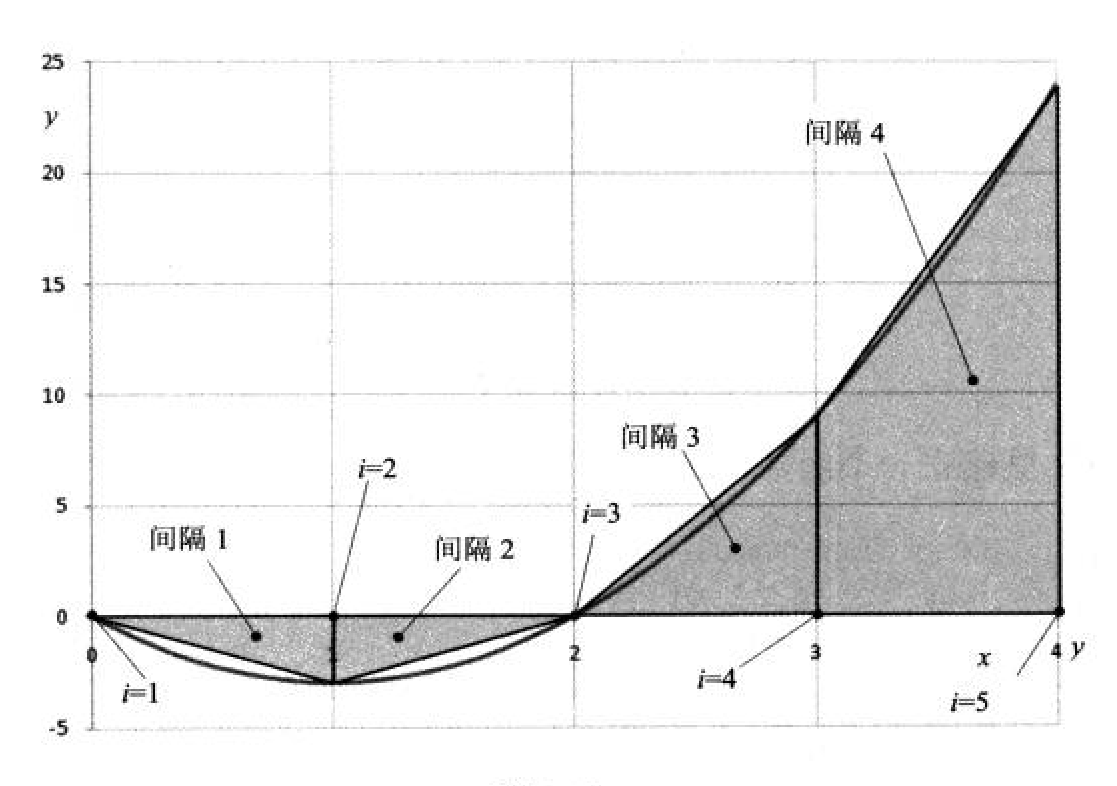


图 9-18

现在我们输入 k 的不同值，重新求这个函数的值。

在命令窗口中，输入不同的间隔数，求这个函数的值。

```
>> intexample(1)
ans =
     48

>> intexample(2)
ans =
     24

>> intexample(8)
ans =
     16.5000

>> intexample(12)
ans =
     16.2222
```

```
>> intexample(20)
ans =
      16.0800

>> intexample(100)
ans =
      16.0032

>> intexample(1000)
ans =
      16.0000
```

注意，当 k=1 时，整个曲线用一个三角形表示，三角形的底边为 4 个单位，高为 24 个单位。随着步数的增加，结果越来越趋近于 16，我们知道它就是这个积分的准确值。

如果我们想进一步增加步数，则需要显示更多的小数位数。输入 format long 命令，它将显示 MATLAB 保存的全部小数位数，重新计算间隔为 10 000 和 50 000 的面积值。

```
>> format long

>> intexample(10000)
ans =
   16.000000319990953

>> intexample(50000)
ans =
   16.000000012868973
```

在命令窗口输入 format short 命令，把显示格式改为短格式，即只显示 4 位小数。

需要说明的是，在中等速度的 PC 上，执行上述的 10 000 次循环，MATLAB 只需要一秒或两秒时间。记住，如果读者输入一个非常大的数，而不想等待输出全部结果，可以按 Ctrl+C 组合键终止输出过程。

例 9.3

用梯形法求下面积分的近似值：

$$\int_{-\infty}^{z}\frac{1}{\sqrt{2\pi}}e^{-Z^2/2}dZ \tag{9-7}$$

式中 Z 是相对于正态分布平均值的偏差数。

如果有 $\bar{x}$ 表示正态分布的平均值，σ 表示标准偏差，则某一个值 x 的 Z 值由下面的公式确定：

$$Z=\frac{x-\bar{x}}{\sigma} \tag{9-8}$$

解：

在开始用梯形法求积分值之前，我们有必要解释一下这个积分公式的含义。其中被积函数：

$$\frac{1}{\sqrt{2\pi}}e^{-z^2/2}$$

被定义为一个正态分布，也称为钟形曲线，如图 9-19 所示。

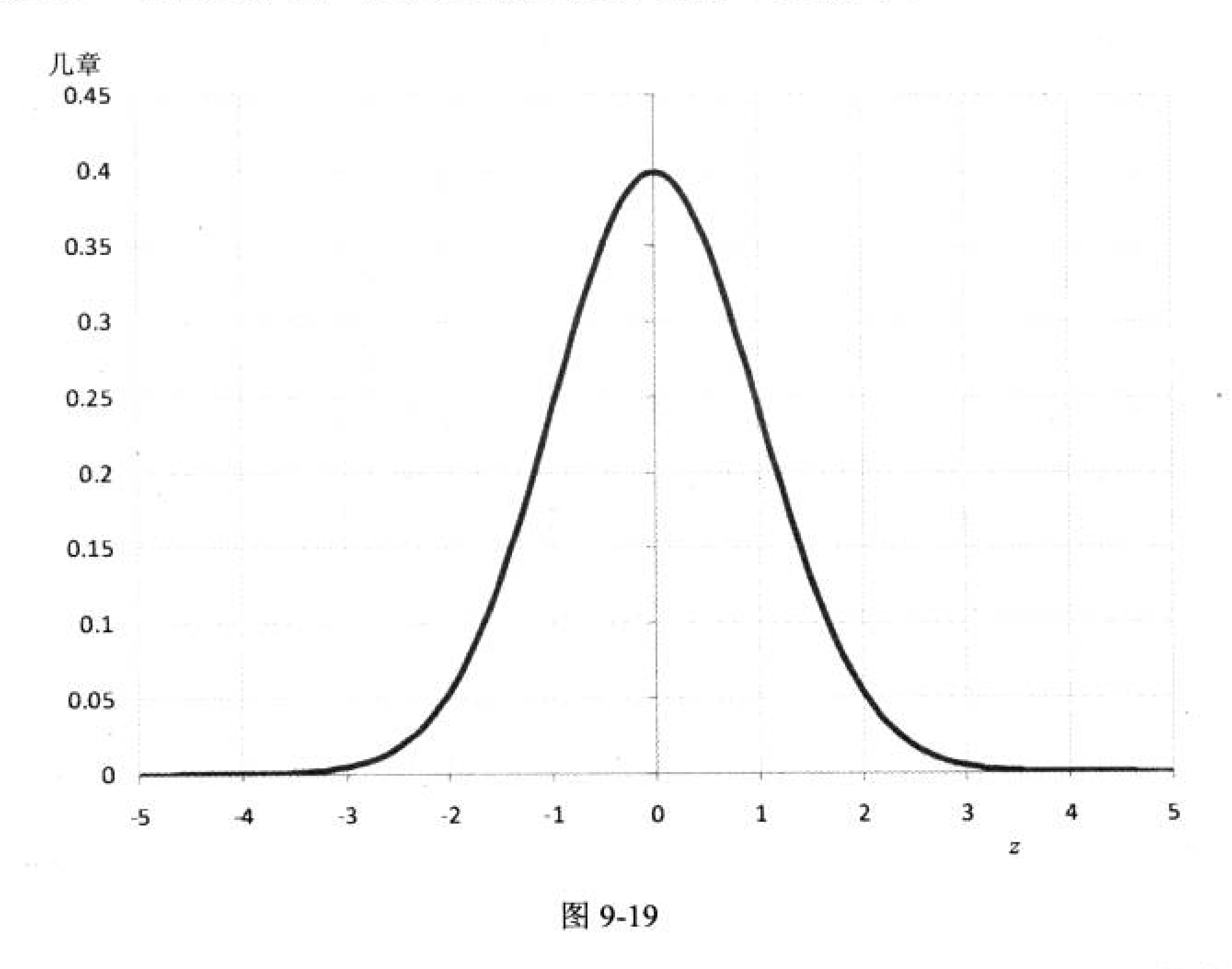


图 9-19

在统计领域及可靠性和质量控制等许多工程应用中，正态分布函数有重要的应用。例如，对于一个制造过程而言，某个公司生产直径为 2in 的钢管。任何制造过程都不可能是完美无瑕的。因此，实际生产的钢管的直径可能会稍微不同于 2in 这个理想值。经过对大量的钢管(样本)进行测试后，我们得到一个平均直径。此外，标准偏差是对实际直径值的变异程度的衡量。大多数的计算器、Excel 或 MATLAB 都有求标准偏差的功能。如果认为该样本代表所有制造的钢管(分布)，则该分布符合正态分布。我们可以根据正态分布的一些性质得到一些重要的结果。例如，如果钢管直径的平均值为 1.990in，标准偏差为 0.006，我们再进一步进行假设，制造厂家保证生产的钢管直径在 1.990in~2.010in，那么生产的钢管处于此范围的百分比为多大？

为了回答这个问题，分析图 9-19 所示的正态分布曲线。曲线围成的整个面积为 1(100%全部分布)，但是这个曲线的两端伸展到正负无穷远，如何计算负穷远的积分呢？答案是选择一个合理的下限值，保证能得到比较准确的结果。假如要求结果准确到小数点后第 6 位小数，那么我们需要反复测试各个不同的下限值(即有多少个标准偏差在这个下限之外)和不同的间隔大小。

建立一个 MATLAB 函数，输入以下的代码：

```
1   function SUM = normdist(limit, k)
2   lower = -limit;
3   upper = limit;
```

```
4    inc = (upper-lower)/k;
5    SUM = 0;
6    x(1) = lower;
7    y(1) = 1/sqrt(2*pi)*exp(-x(1)^2/2);
8    for i = 2:(k+1)
9       x(i) = x(i-1)+inc;
10      y(i) = 1/sqrt(2*pi)*exp(-x(i)^2/2);
11      SUM = SUM + .5*(y(i) + y(i-1))*(x(i) - x(i-1));
12   end
```

把这个脚本程序保存为 normdist。输入 *limit* 和 *k* 不同的值，反复测试这个函数。

```
>> format long

>> normdist(3,100)
ans =
   0.997292229481189

>> normdist(3,1000)
ans =
   0.997300124163755

>> normdist(6,1000)
ans =
   0.999999998025951

>> normdist(6,10000)
ans =
   0.999999998026819

>> normdist(6,100)
ans =
   0.999999997940018

>> normdist(5,100)
ans =
   0.999999414352763
```

我们注意到，改变偏差相对值对准确度有很大的影响。而改变间隔数的影响比较少。例如用±6 个偏差，即使计算区间间隔只有 100 个，得到的结果精确到小数位 8 位。因此，我们选择-6 为下限值(即低于平均值的 6 个标准偏差)，间隔数为 1000。由于即使把间隔数从 100 改为 1000，也不会引起计算时间显著变化。现在我们修改这个 MATLAB 函数，根据 *Z* 的下限值求这个被积函数的积分值。

根据下面的要求对 normdist( )函数进行修改(注意第 6 行~第 14 行没有变化)：

```
1    function SUM = normdist(Z)
2    % Calculates the % of population with values less than Z
3    lower = -6;
4    upper = Z;
5    k = 1000;
```

```
6   inc = (upper-lower)/k;
7   SUM = 0;
8   x(1) = lower;
9   y(1) = 1/sqrt(2*pi)*exp(-x(1)^2/2);
10  for i = 2:(k+1)
11       x(i) = x(i-1)+inc;
12       y(i) = 1/sqrt(2*pi)*exp(-x(i)^2/2);
13       SUM = SUM + .5*(y(i) + y(i-1))*(x(i) -x(i-1));
14 end
```

保存修改后的程序。把数值显示格式设置为短格式，分别输入 Z=0 和 Z=1，执行这个函数：

```
>> normdist(0)
ans =
     0.5000
>> normdist(1)
ans =
     0.8412
```

Z=0 时，积分的值为 0.5，这个结果是符合情理的。因为有 50%的测量值小于平均值(表示相对平均值的 0 个标准偏差)。Z=1 的结果，可以从图 9-20 得到解释。正态分布曲线在相对于平均值 1 个标准偏差之下围成的面积是 0.841 2。因此有 84%的分布落在此范围内。

我们已经建立了一个可以计算正态分布积分值的函数，现在返回到刚开始提出的问题。

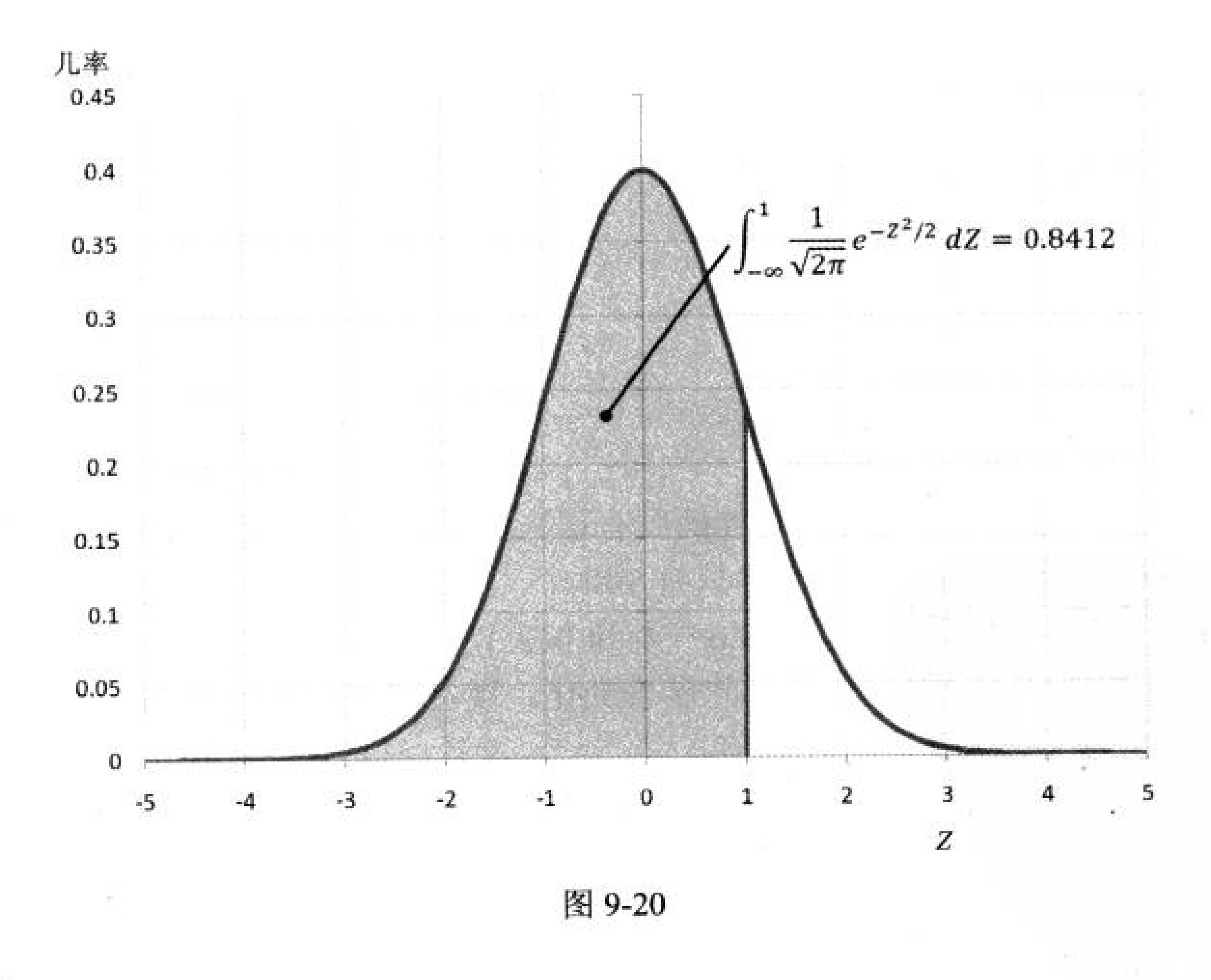


图 9-20

例 9.4

如果制造的钢管分布曲线的平均直径是 1.998in，标准偏差为 0.006，那么生产的钢管

直径落在 1.990in~2.010in 范围内的百分比为多少？

解：

首先求 Z 的下限值：

$$Z = \frac{1.990 - 1.998}{0.006} = -1.333 \tag{9-9}$$

即下限值为 1.333 个标准偏差。

把 Z= − 1.333 代入前面的 normdist( )函数：

```
>> normdist(-1.333)
ans =
     0.0913
```

这个结果可以这样来解释：全部钢管中有 9.13%的直径小于这个下限值。

现在我们来求上限值：

$$Z = \frac{2.010 - 1.998}{0.006} = 2.000 \tag{9-10}$$

因此，此上限值表示 2 个标准偏差。

把 Z=2.0 代入 mormdist( )函数：

```
>> normdist(2)
ans =
     0.9772
```

从这个结果可知，全部钢管中有 97.72%的钢管的直径小于这个上限值，而有 2.28%的钢管的直径大于这个上限值。因此保证在合格范围内的百分比是：

97.72-9.13=88.59%

注意，我们也可以用在一行内两次调用这个函数，这样一步就可以解决这个问题。

```
>> normdist(2) - normdist(-1.333)
ans =
       0.8860
```

结果与原来的值有微小的差异。这是由在计算过程对中间值舍入的原因引起的。图 9-21 直观地说明了这个结果。对于大多数制造工厂，这个结果是不可接受的。因为有 11%的钢管不能满足公司的规格要求。必须采取措施改进生产过程，或者由公司修改产品规格。

在这个例子里，我们用数值积分法计算一个积分公式，这个积分公式没有一个解析解。但由于它在统计中的重要性，许多参考书专门用表格列出了 Z 值和相应的积分值。我们也称这个积分为累积密度函数。Excel 有许多内置的统计函数，如 NORMSDIST( )，它就是根据 Z 值计算累积密度函数的值。

为了验证 MATLAB 里的运算结果，我们在任意一个单元格里输入以下公式：

=NORMSDIST(2) - NORMSDIST(-1.333)

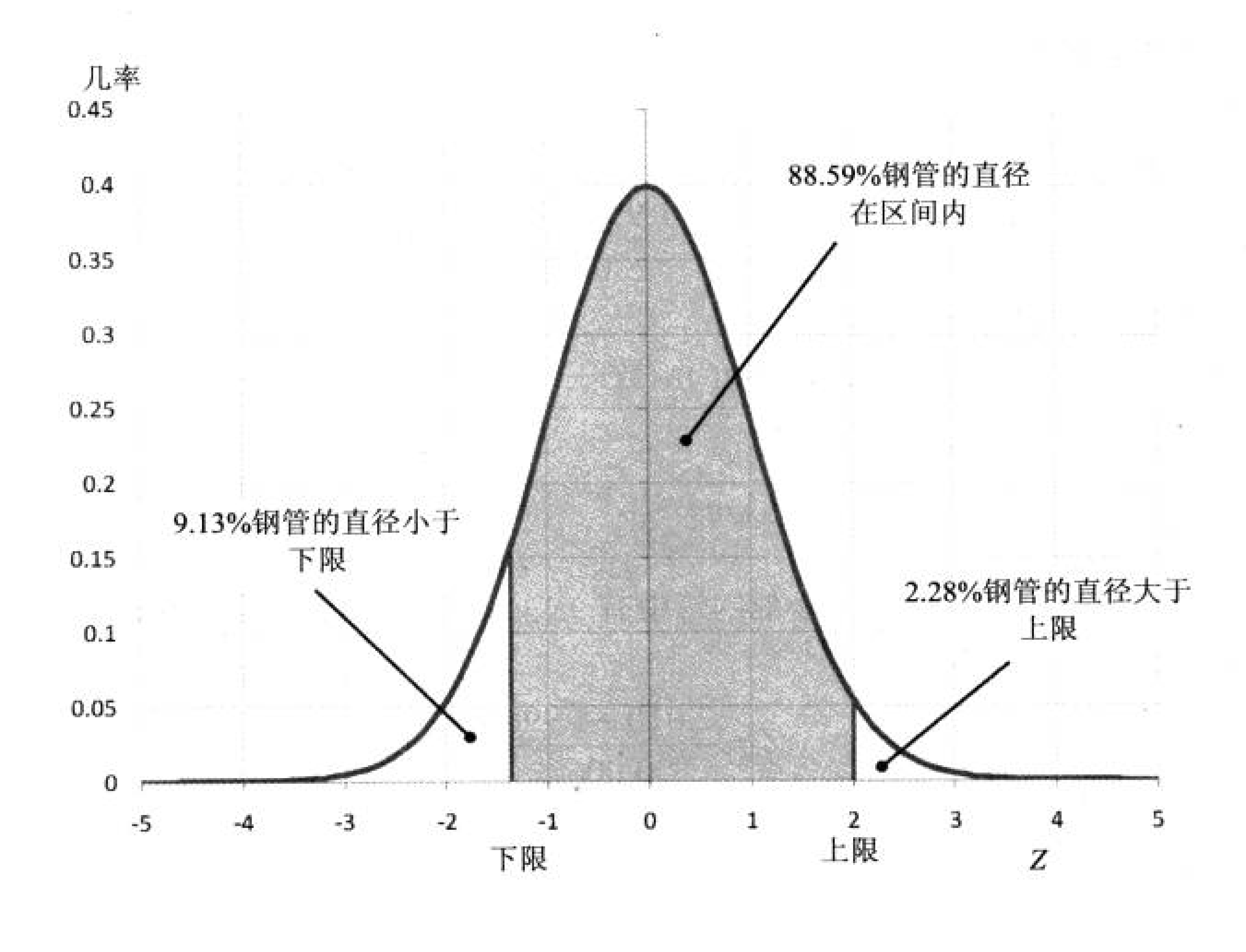


图 9-21

9.3 测量数据的数值积分

在前一节里，我们给定了积分的函数。但是还经常遇到另一种情况。事先并不知道需要积分的函数，它是由一组测量数据确定的。在下面的这个例子里，我们对一组数据点进行积分，求出速度和位置值。

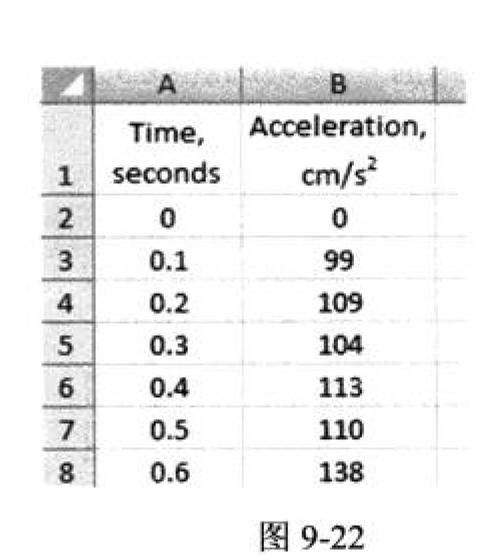
例 9.5

一个小机器人沿着一条直线移动。机器上装有一个加速度测量仪，可以每秒 10 次记录沿着运行方向(x 轴)的加速度。表 9.1 是该测量仪在 4s 内记录下来的数据。假设在时间 t=0 时刻，该机器人处于静止状态(速度和加速度都为 0)。求出该机器人的最大速度，在 4s 时间结束时的速度和经过的总路程。

解：

打开一个电子表格，根据图 9-22，在 A 列里输入时间值(从 t=0 开始)，在 B 列里输入加速度值。注意，为了使单元格内的一行文字自动换行，如行 1 所示，从 Ribbon 的开始组选择【自动换行】工具，或者按下 Alt+Enter 组合键，强制插入一个换行符号。

绘制加速度与时间的曲线，如图 9-23 所示。



	A	B
1	Time, seconds	Acceleration, cm/s²
2	0	0
3	0.1	99
4	0.2	109
5	0.3	104
6	0.4	113
7	0.5	110
8	0.6	138

图 9-22

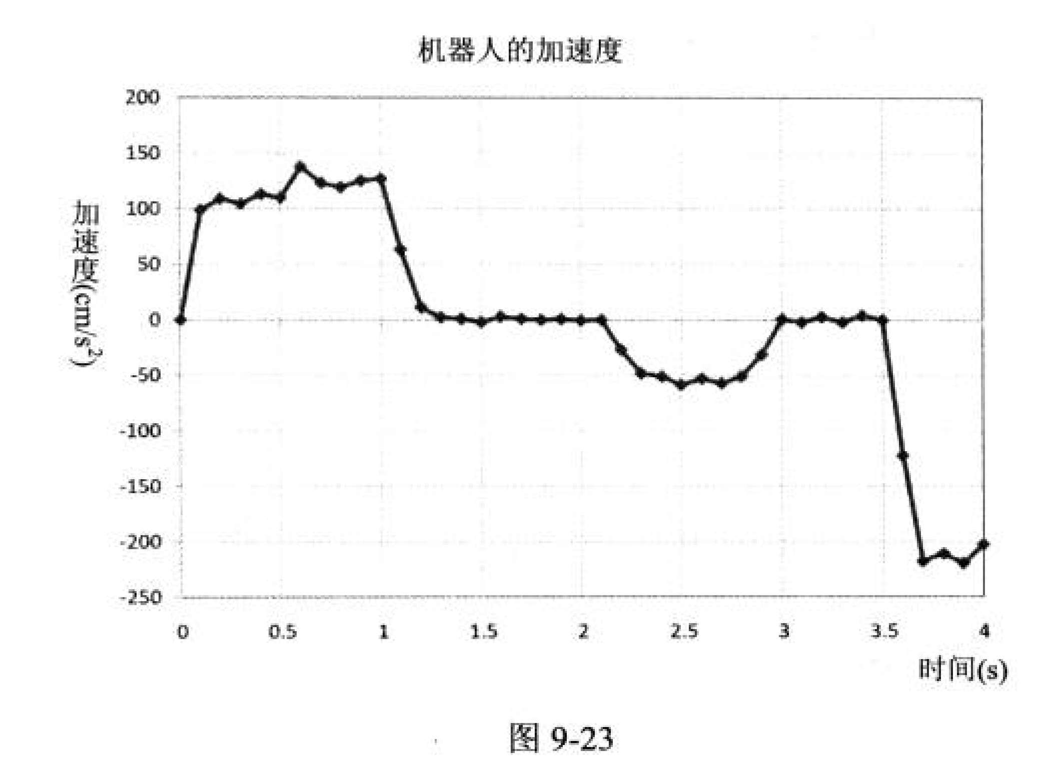


图 9-23

表 9-1　机器人的加速度数据

时间(s)	加速度 (cm/s²)	时间(s)	加速度 (cm/s²)	时间(s)	加速度 (cm/s²)	时间(s)	加速度 (cm/s²)
0.1	99	1.1	63	2.1	0	3.1	−2
0.2	109	1.2	11	2.2	−27	3.2	3
0.3	104	1.3	2	2.3	−48	3.3	−2
0.4	113	1.4	1	2.4	−51	3.4	4
0.5	110	1.5	−2	2.5	−58	3.5	0
0.6	138	1.6	3	2.6	−53	3.6	−122
0.7	123	1.7	1	2.7	−57	3.7	−218
0.8	119	1.8	0	2.8	−51	3.8	−211
0.9	125	1.9	1	2.9	−31	3.9	−220
1.0	127	2.0	−1	3.0	1	4.0	−203

分析机器人的运动曲线，我们可以看出，该机器人在第一秒时间内，速度逐渐增加。在第二秒时间内，加速度接近于 0，因此速度基本保持不变。在第 3 秒和第 4 秒时间内，先是一段减速运动，紧跟半分钟的匀速运动，最后的半分钟是急剧的减速度运动。

我们重复前面的方程(9-2)，速度是加速度的积分：

$$v = \int a dt \tag{9-2}$$

因此，在加速度曲线下每个间隔内的面积是该区间对前一时刻的速度变化值。由于机器人的初始速度为 0(初始状态下)，因此最终速度等于这个曲线以下的面积。

在 C 列里，输入计算第一个梯形面积的公式，如图 9-24 所示。注意，这个值必须输入在与 t=0.1s 相对应的行里，计算得到的面积表示速度从 t=0s 到 t=0.1s 的变化。面积的单

位就是梯形底边的单位与高的单位的乘积，即：

$$(\text{second})\left(\frac{\text{cm}}{\text{second}^2}\right)=\frac{\text{cm}}{\text{second}}$$

把这个单元格里的公式复制到 C 列其他单元格。在 D 列，在与 t=0 对应的行里，输入初速度 0。在下一个单元里输入一个公式，它把速度变化值加到前一时刻的速度上，如图 9-25 所示。把这个公式复制到 D 列的其他单元格。

	A	B	C	D
1	Time, seconds	Acceleration, cm/s²	Velocity Change, cm/s	
2	0	0		
3	0.1	99	=(B3+B2)/2*(A3-A2)	
4	0.2	109		
5	0.3	104		

图 9-24

	A	B	C	D
1	Time, seconds	Acceleration, cm/s²	Velocity Change, cm/s	Velocity, cm/s
2	0	0		0
3	0.1	99	4.95	=D2+C3
4	0.2	109	10.4	
5	0.3	104	10.65	

图 9-25

绘制速度对时间的曲线，如图 9-26 所示。

从图 9-26 可以看出，最大速度大概是 125cm/s。如果读者要得到一个更加准确的结果，可以在电子表格里向下翻页，找到最大值的位置，但是一个更快的办法是输入求整列数据最大值的公式。当数据量多于几百个或甚至几千个时，这种方法特别有用。

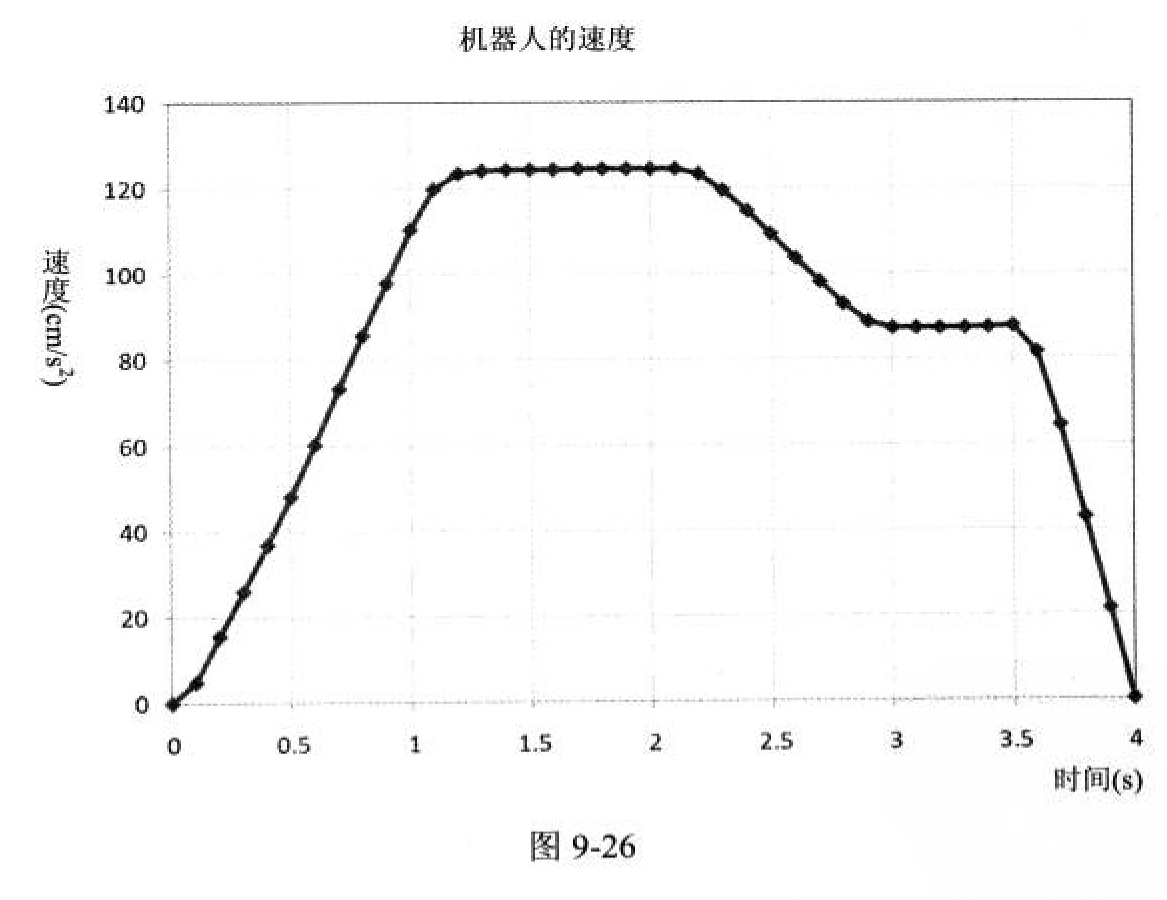


图 9-26

在 H 列里输入最大值公式，求出 D 列的最大值，如图 9-27 所示(保留 E 和 F 列用于位置计算)。单击列标头，选取整列数据，就会看到最大速度值，如图 9-28 所示。

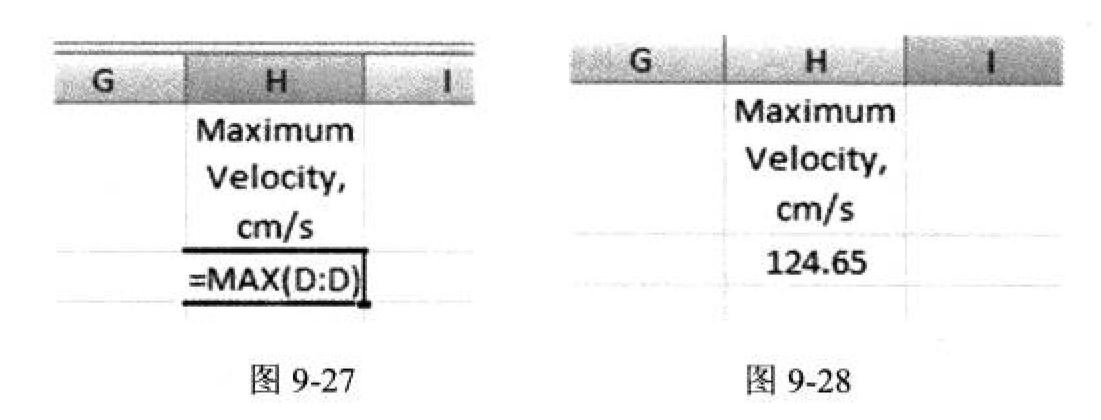


图 9-27　　　　图 9-28

我们还发现最终的速度接近于 0。

在 x 轴的位置变化等于速度的积分。我们设置初始位置在 x=0 位置。

在 E 列，输入一个公式求速度对第一个时间间隔的积分，如图 9-29 所示，把这个公式复制到同一列的其他单元格。注意现在的单位是

$$(\text{second})\left(\frac{\text{cm}}{\text{second}}\right)=\text{cm}$$

	A	B	C	D	E	F
1	Time, seconds	Acceleration, cm/s²	Velocity Change, cm/s	Velocity, cm/s	Position Change, cm	
2	0	0		0		
3	0.1	99	4.95	4.95	=(D3+D2)/2*(A3-A2)	
4	0.2	109	10.4	15.35		
5	0.3	104	10.65	26		

图 9-29

在 F 列里，输入机器人的位置公式，它等于前一单元格时的值加上速度变化值，假设初始位置为 0。图 9-30 只显示这个表的其中几行内容。

	A	B	C	D	E	F
1	Time, seconds	Acceleration, cm/s²	Velocity Change, cm/s	Velocity, cm/s	Position Change, cm	Position, cm
2	0	0		0		0
3	0.1	99	4.95	4.95	0.2475	0.2475
4	0.2	109	10.4	15.35	1.015	1.2625
5	0.3	104	10.65	26	2.0675	3.33
6	0.4	113	10.85	36.85	3.1425	6.4725
7	0.5	110	11.15	48	4.2425	10.715
8	0.6	138	12.4	60.4	5.42	16.135
9	0.7	123	13.05	73.45	6.6925	22.8275
10	0.8	119	12.1	85.55	7.95	30.7775

图 9-30

绘制位移曲线，如图 9-31 所示。

最后的位移，350cm，即 3.5m 就是机器人在 4s 内经过的距离。

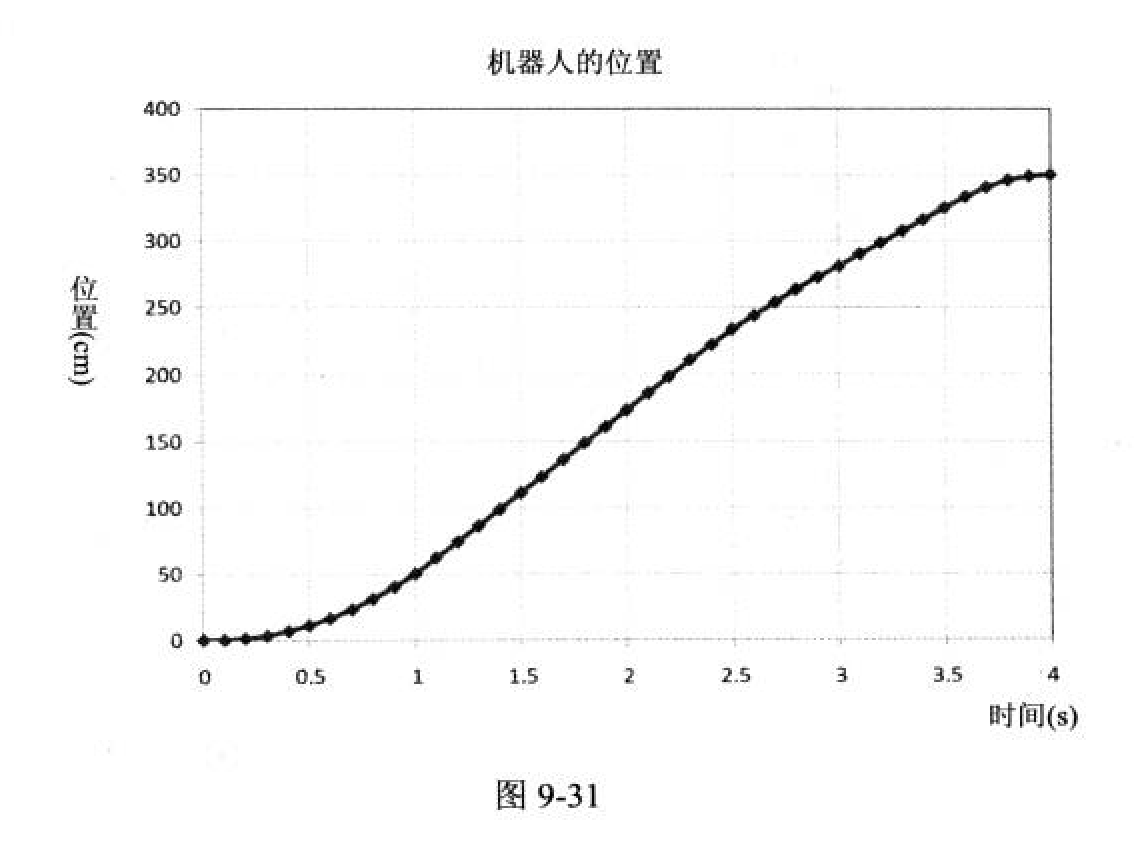


图 9-31

9.4 习题

1. 用 Excel 画出以下函数的曲线：

$$y = x^2 - 3x + 3 \qquad 0 \leqslant x \leqslant 4$$

用手工制作一个表格列出 y 在 x 整数值位置。用直线段把这些点连接起来，并计算每个梯形的面积。将这些面积相加，把结果与根据下面的积分公式求得的准确结果相比较。

$$\int_0^4 (x^2 - 3x + 3)dx = \left(\frac{x^3}{3} - \frac{3x^2}{2} + 3x\right)\Bigg|_0^4 = 9.3\overline{3}$$

2. 用 Excel 求出以下积分公式的数值结果，间隔数分别为 2、4、8 和 16。对结果加以解释。

(a) $\int_0^{\pi} \sin\theta d\theta$

(b) $\int_0^{2} e^x dx$

(c) $\int_0^{4} (x^3 + 3x^2 - 20)dx$

3. 编写一个 MATLAB m-file，计算下面的积分值。

$$\int_0^4 4^{-x}\,dx$$

4. 修改习题 3 的 m-file，重新计算积分公式，把上限值分别取 4、6、8 和 10。解释得到的结果。

5. 利用例 9.3 的 normdist( )函数计算以下问题：钢质螺钉分布曲线的平均断裂强度为 74 000bl，标准偏差为 2 000bl。一个螺钉的断裂强度小于 67 000bl 的机率为多大？

6. 利用例 9.3 的 normdist( )函数计算以下问题：假设美国成年男子的平均身高是 69in(5ft，9in)，标准偏差为 3.0in。假设身高呈正态分布，求出符合以下要求的美国成年男人的百分比：

(a) 至少 6ft

(b) 小于 5ft

(c) 大于 7ft

(d) 5ft6in~6ft

7. 一个火箭模型从地面发身到天空。火箭的发动机燃烧 3s，4.5s 之后燃料完全烧完。加速度测量仪每隔 0.5s 记录火箭的加速度，测量值如表 9-2 所示。其中，正值表示向上。当燃料完全烧完后，加速度为-g，即-9.8m/s^2，这是因为它受到重力加速度的作用。用 Excel 求：

(a) 最大的向上速度

(b) 飞行的最大高度

(c) 火箭落地的时间

(d) 火箭落地时的速度

除此之外，分别在 3 张不同的图上，绘制加速度、速度和高度随时间变化的曲线 。注意：一定计算到火箭落地为止。

表 9-2 火箭的加速度表

时间(s)	加速度(m/s^2)
0	0
0.5	30.1
1	32.4
1.5	33.6
2	35.1
2.5	38.7
3	34.2
3.5	18.1
4	–2.4
4.5	–9.8
5	–9.8

在剩下的飞行时间里 $a=-9.8m/s^2$

8. 在例 9.5 机器人问题里，假如质量为 1.5kg。根据下列要求用 Excel 求机器的电机在前 1.4s 内所做的功：

(a) 用牛顿定律(F＝ma)，机器人的质量和表 9.1 的加速度求机器人的电机在 1.4s 之前每个数据点上产生的推力。

(b) 根据推力数据和位置数据(由例 9.5 得到)绘制推机器人在前 1.4s 时间内力与位移的关系曲线。

(c) 功可以按下面的公式计算：

$$W\int Fdx$$

式中，F 是作用力，x 是经过的距离。利用 b 的曲线和数值积分法求机器人电机所做的功。要说明答案的单位。

9. 在第 5 章里，我们绘制了弹簧振子振动曲线。我们还经常需要分析系统对受迫振动的响应。这个振动系统可用图 9-32 表示。

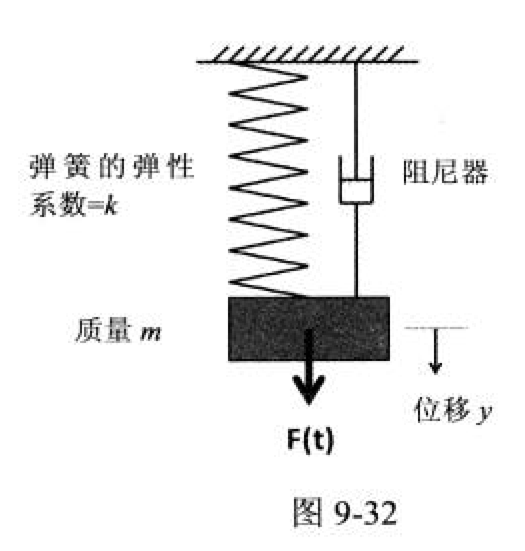


图 9-32

在任意时刻 t_1 的位移 y 由下面的 Duhamel 积分公式决定：

$$y(t_1)=\frac{1}{m\omega_D}\int_0^{t_1}F(t)e^{-\xi\omega(t_1-t)}\sin([\omega_D(t_1-t)]dx$$

式中：

$\omega=\sqrt{\dfrac{k}{m}}$ 是固有频率。

$\omega_D=\omega\sqrt{1-\xi^2}$ 是受阻尼的固有频率。

ξ 是阻尼系数，它从 0(无阻尼)变化到 1(临界阻尼)。

$F(t)$是外力函数(外力随时间变化)。

设计一个 MATLAB 函数，计算系统在任意时间 t_1 的位移，要求输入参数有：

m=1kg，k=100N/m，ξ＝0.1

作用力函数为：

F＝9.8N

这里的作用力只是物体的重力。这个问题的物理解释是：初始时，振子处于静止状态，它的位置就是弹簧不受外力时的长度。在时间 t=0 时，释放振子，因此同时受到重力的作用。

时间从 0 开始经过 1000 步后到达 t_1。考虑到外力是时间的函数，因此另外设计一个 MATLAB 函数表示外力。这样处理的好处是，在以后的问题中可以方便修改外力。

用下面的值进行验证：

$y(t_1=1\text{s})=0.131\text{m}$

$y(t_1=5\text{s})=0.097\text{m}$

注意，随着时间的递增，位移越来越接近于平衡位置。这个平衡位置由振子的重量除以弹簧的弹性系数得到。

10. 编写一个 MATLAB 程序，利用习题 9 的函数，求 0s~5s 时间内位移的变化情况，并绘制它的曲线。时间间隔为 1000 步。并对曲线的形状加以解释。分别对阻尼系数为 0(无阻尼)和 0.5 两种情况，重复此计算过程。

11. 重复习题 10，假设外力不是在瞬间施加于物体，而是在前 2s 时间内线性递增，如图 9-33 所示。解释计算结果。

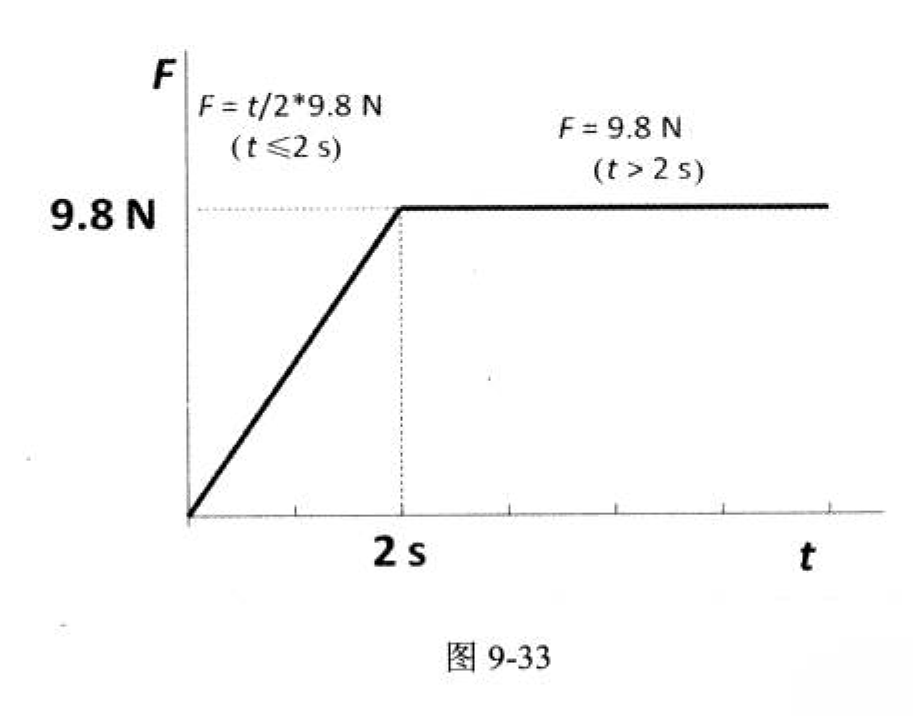


图 9-33

12. 根据以下的外力函数，重复习题 11，假设阻尼系数为 0：

(a) F=10sin(5t)

(b) F=10sin(10t)

(c) F=10sin(15t)

在问题 b 里外力的频率等于系统的固有频率，读者会注意到什么现象呢？如果读者曾经驾驶过一辆车轮严重不平衡的汽车，则可能已经注意到共振的条件，当汽车达到某个速度时，汽车震动得特别厉害，在这个速度，车轮频率与汽车的固有频率大致相等。

13. 一个 400gal 的圆柱桶储满了水。水桶底部的阀门打开后，流过这个阀门的流速 q(每分钟加仑)为：

$$q(t) = 80e^{-0.2t}$$

式中，t 是时间，单位为 min。流过这个阀门的总水量等于这个函数对时间的积分。用 MATLAB 求：

(a) 这个水桶全部的水流掉一半所需要的时间。

(b) 15 分钟后，留在桶里的水有多少？

14. 飞机机翼的横截面是一个螺旋桨形状。设计螺旋桨需要了解空气动力学的高深知识。但是，现在飞机在设计时有很多标准的螺旋桨供设计人员利用。这些标准的螺旋桨通常用无量纲的 x-y 坐标系来描述。如螺旋桨的某一点在(0,0)位置，尾端在(1,0)位置，等等。

找到并下载 Selig 1223 螺旋桨的坐标点。这方面一个非常准确的数据是在 UIUC 螺旋桨坐标数据库(可以在 http://www.ae.uiuc.edu/m-selig/ads/coord_database.html 网站上找到这个数据库)。虽然坐标是无量纲的，而且可以缩放到任意的单位系统，但是我们这里假定坐标单位为 m。

(a) 用 Excel 绘制螺旋桨的坐标图，看看它们是什么形状(必须同时放大或缩小 x 和 y 坐标轴，否则不能得到正确的螺旋桨形状)。

(b) 在 Excel 里用数值计算求螺旋桨的横截面积。

第10章

最优化

引言

工程师的主要任务之一就是设计新设备、工艺和系统。工程师的设计过程定义为：

“把科学和物理原理应用于确定一个能满足一定性能要求，在某些方面具有最佳性能的设备、工艺或系统的过程”。

工程师的目标就是找出问题的最佳解。这个思想非常重要，这表明，工程师不仅要找出问题的答案，而且这个答案根据某些客观标准是最佳答案。工程师们用来确定哪个答案是最佳答案的准则则因问题而异，也因人而异。在一些工程问题中，最佳可能意味着成本最低，或者最可靠、体积最小、重量最轻等。在另一些问题中，可能是其他一些准则，或者是几个准则的组合。在设计过程中，工程师们努力寻求使某个属性最大化或最小化的解决方案。因此，最优化理论是工程师们设计的一个重要工具。

最优化是确定一个数学函数的最大值或最小值的数学过程，或者确定使函数达到最大值或最小值的自变量的值。一些非常简单的情况只有几个自变量，而且自变量的取值范围也不受限制。微积分是经常用来手工求最优值的数学工具。当自变量的个数很多时，或者当自变量受到许多约束时，必须采用算法解才能找到最优答案。本章将介绍如何用数学公式描述最优化问题，还将介绍如何用 MATLAB 和 Excel 里内置的工具来求最优化问题的数值解。

我们将学习以下几方面内容：

- 如何用标准公式描述一个优化问题。
- 线性优化问题与非线性优化问题的差别。
- 如何在 MATLAB 里用 fminsearch( )和 fminbnd( )函数求解最优化问题。
- 如何在 Excel 里用规划求解工具解决最优化问题。

10.1　工程中的最优化问题

为了把数学中的最优化方法应用于工程设计，我们将介绍一个典型的工程设计问题。一位工程师需要设计这样一根横梁：横梁长为 10ft，一端固定在墙上，要求另一端能支持 500bl 的物体，而且位置偏移不能超过 1in。图 10-1 是它的示意图。

仔细分析了设计任务后，我们对设计方案做出如下初步构想：

- 利用圆柱管，圆柱管的横截面各方向的刚度完全一样，这样安装横梁时，不需要按特定朝向安装。
- 为了减轻自身的重量，采用中空的圆管。
- 这个圆管由钢材制造，因为钢材受力强、价格便宜，而且容易得到。

经过这些初始设计的决策后，问题归结为如何选择钢管横截面的内外直径，如图 10-2 所示。

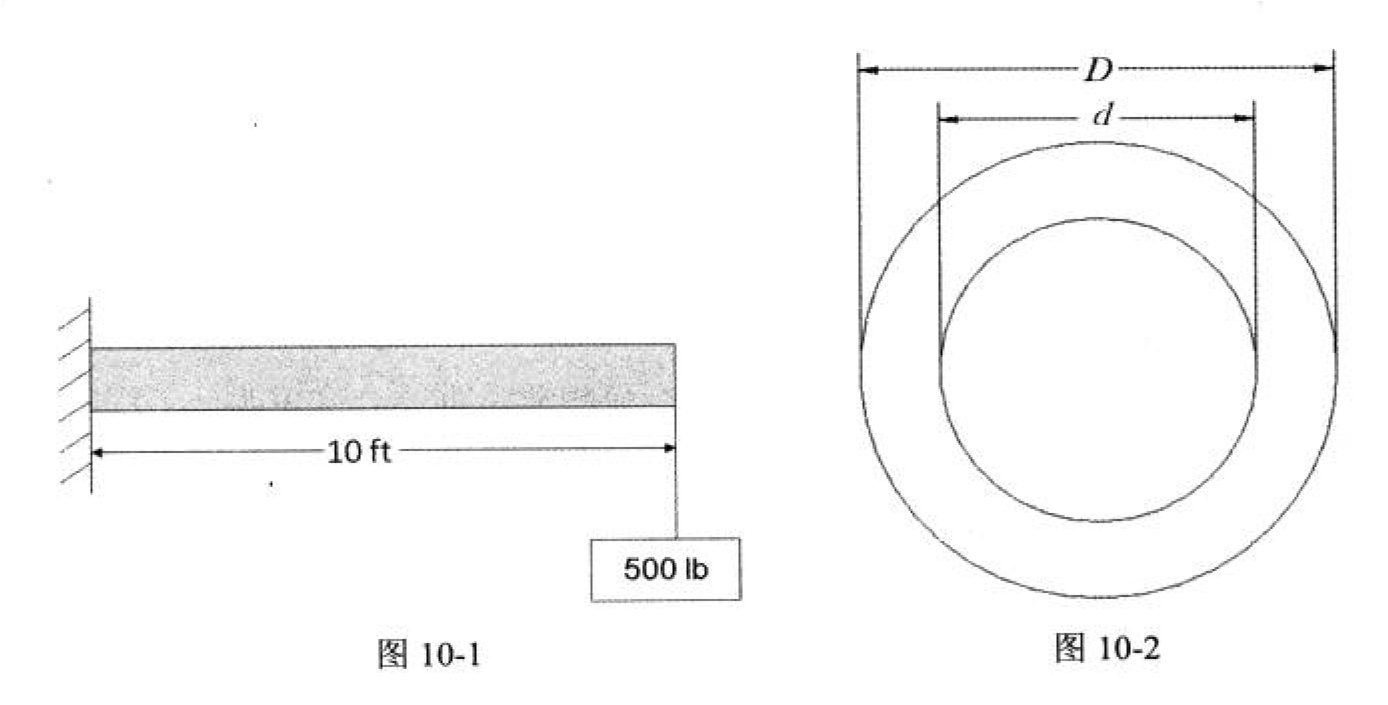


图 10-1　　　　图 10-2

根据材料力学的知识，还需要知道这个横梁的一端在 500lb 重力作用下的偏移值：

$$\delta = \frac{64Wl^3}{3\pi E(D^4 - d^4)} \tag{10-1}$$

式中 δ 是悬挂物体一端的偏移值(单位为 in)，W 是物体的重量(500lb)，l 是横梁的长度(120 in)，E 是材料的弹性模量(钢的弹性模量是 30 000 000lb/ in^2)，D 是外层的直径，d 是内层的直径，单位都是 in。由于设计问题，允许 1in 的偏移，我们把上述参数代入方程(10-1)得到以下的方程：

$$\frac{64Wl^3}{3\pi E(D^4 - d^4)} \le 1in \tag{10-2}$$

分析这个方程后，我们发现有两个未知量，它们是 D 和 d。因此称这个问题为欠约束问题，这会使得此问题可能会有无数个解。在这种情况下，可以把它当作“自由选择”的机会。我们可以将其中一个未知量给定任意值，再利用方程(10-2)求出另一个未知量的值。

例如，假设 d=1in，则利用方程(10-2)得到如下的结果

$$D^4 \geq \frac{64(500\text{lb})(120\text{in})^3}{3\pi(30000000)(1\,\text{in})} + d^4 \tag{10-3}$$

计算结果得到 D≥3.74。根据这个运算结果，我们向钢材供应商咨询，选择一个可用的标准的钢管，它的外直径要大于 3.74in(为了保险起见，可能选择$3\frac{3}{4}$ in，或$3\frac{7}{8}$ in)，这完全是满足这个问题要求的一个可接受的答案。

但是作为工程师，不能只寻求一个可接受的答案，而是要根据某个准则得到一个最优的解。当我们面对无数的自由选择时，并不是任意地给某个设计变量设置一个值，而是精心考虑全部变量的值，使得它们的组合是一个最优的答案。这就是优化设计。

在我们这个例子里，最优化的解决方案可能是用钢材量最少的方案。因此，为了计算这个钢管的重量，需要计算它的体积，下面是这个钢管的体积公式：

$$v = \frac{l\pi}{4}(D^2 - d^2) \tag{10-4}$$

我们就是根据这个准则，判断解决方案的优劣。并从多个设计方案中选择钢材量最小的方案(最小化)。按最优化的术语，我们称它为目标函数。

从数学角度来分析方程(10-4)，当 D 与 d 相等时，V 达到最小值 0。然而在我们这个例子里，对变量的取值有若干个物理约束。例如，横梁一端的偏移不能超过 1in。因此，必须满足方程(10-2)，除此之外，还有其他一些约束条件：

- D 和 d 必须是正的非零值，因为它们表示实际的物理量。
- 从制造角度来看，制造一个非常薄的钢管是非常难的，因此我们把钢管壁的厚度至少定为 0.125in。

按照最优化的术语，我们称上述对变量取值的 4 个限制为约束。根据最优化理论，我们把这些方程写成标准形式，即写成等式或不等式的形式，所有的自变量出现在方程的左侧。并把这个问题的目标函数和约束条件写成最优化的标准语言：

最小值：

$$v = \frac{l\pi}{4}(D^2 - d^2)$$

受约束条件：

$$\frac{64Wl^3}{3\pi E(D^4 - d^4)} \leq 1in$$

$$\frac{D-d}{2} \geq 0.125\text{in}$$

$$D \geq 0$$

$$d \geq 0 \tag{10-5}$$

最优的设计方案就是 D 和 d 的值都满足上述的全部约束条件并且使目标函数最小化。

10.2　最优化问题的描述

为了归纳并且用标准语言描述这个例子用到的步骤，必须按照下面的步骤用最优化理论的语言描述最优化问题。

- 步骤 1：确定需要赋值的自变量。在工程最优化问题里，我们称这些变量为设计变量。可能会有 n 个这样的变量，我们用 x_1，x_2，x_3，…，x_n 表示这些变量。
- 步骤 2：确定目标函数，或在最优化过程中需要得到最小值或最大值的量。这正是用来判断“最优”解的准则：

$$y=f(x_1,x_2,x_3,\ldots,x_n) \tag{10-6}$$

- 步骤 3：确定作用于自(设计)变量上的限制条件，把它们写成等式或不等式的形式。我们称这些限制为约束条件。每个约束条件都要使用下列形式之一：

$$g(x_1,x_1,x_3,\cdots,x_n)=A_i \tag{10-7}$$

$$g(x_1,x_1,x_3,\cdots,x_n)\le A_i \tag{10-8}$$

$$g(x_1,x_1,x_3,\cdots,x_n)\ge A_i \tag{10-9}$$

其中 A_i 是一个常量。

步骤 3 需要引起读者的特别注意。全部约束条件都要一一确定，而且都要写成方程的形式。即使有些约束条件非常明显，或者不重要，也必须一一考虑到。例如在第 10.1 节里，我们增加了 D≥0 和 d≥0 这两个约束条件。虽然这些条件对于工程师们来说是非常明显的，但是也必须被考虑到，而且写成方程的形式。如果忽略了这些明显的约束条件，我们最后得到的结果可能会出现直径的负值。此外还要注意，在某些问题里，根本不存在对设计变量的约束条件。然而这在工程问题中非常少见。

一旦我们把最优化问题写成标准形式之后，确定最优化问题的类型是非常有必要的。这会帮助我们选择最优化问题的解决方法。

- **受约束和无约束：**一个不受约束的最优化问题是指对设计变量没有任何约束。因此也就没有约束方程。如果有约束条件作用于设计变量上，则这样的最优化问题被称为受约束最优化问题。
- **线性与非线性：**线性代数方程是指自变量单独出现或者乘上一个常量。下面是线性方程可的形式：

$$f(x_1,x_1,x_3,\cdots,x_n)=a_1x_1+a_2x_2+\cdots+a_nx_n \tag{10-10}$$

式中 a_i 都是常量(包括 0 和 1)。如果目标函数和所有的约束方程都是这样的线性形式，我们把这类问题归为线性最优化问题。如果目标函数或其中任意一个约束方程是一个非线性形式(即方程包括设计变量的指数大于 1 如(x_i^3)，或者包括设计变量的三角函数，如 $\cos(\pi x_2)$)，我们把这一类问题归结为非线性最优化问题。注

意，在约束方程里，方程(10-10)中的等号改为不等号(≤，≥)，不会影响问题的分类。

虽然从严格意义上讲这种分类并不是十分必要的，但是它有助于我们选择求解方法，并且有助于我们理解解决方法的性质。

10.3 最优化问题的求解

本节将分析各类最优化问题的求解方法。虽然只举了几个简单的例子，但是这些例子能帮助我们选择求解的方法，并且帮助我们更好地理解更加复杂问题的解决方法。我们先考虑非线性无约束最优化问题，接着考虑线性约束最优化问题，最后考虑非线性约束最优化问题。

10.3.1 非线性无约束最优化问题

回忆一下第 10.2 节的内容，一个非线性无约束最优化问题由需要最大化或最小化的非线性目标函数组成，设计变量的取值不受任何约束(即没有约束方程)。由于没有约束方程，因此这类问题是最简单的最优化问题。

为了更好地理解这类问题的性质，我们考虑一个简单的例子，它只有一个设计变量 x 和一个非线性目标函数 $f(x)$。求解这一类问题，就要求出 $f(x)$出现极值的位置(极小值和极大值)。在一般情况下，我们在 x 的很大范围里分析 $f(x)$与 x 的关系，图 10-3 是它的一个典型曲线。虽然 x 的值不受任何约束，它可以取 $-\infty\sim+\infty$ 之间的任意值，但是通过分析图中所画的曲线，可以对问题有深入的了解。在这种情况下，需要注意以下的特征：

- 这个函数的曲线有多个波峰和多个波谷。
- 每个波峰或波谷的斜率都为 0。

这些特征都可从图 10-4 中看出。

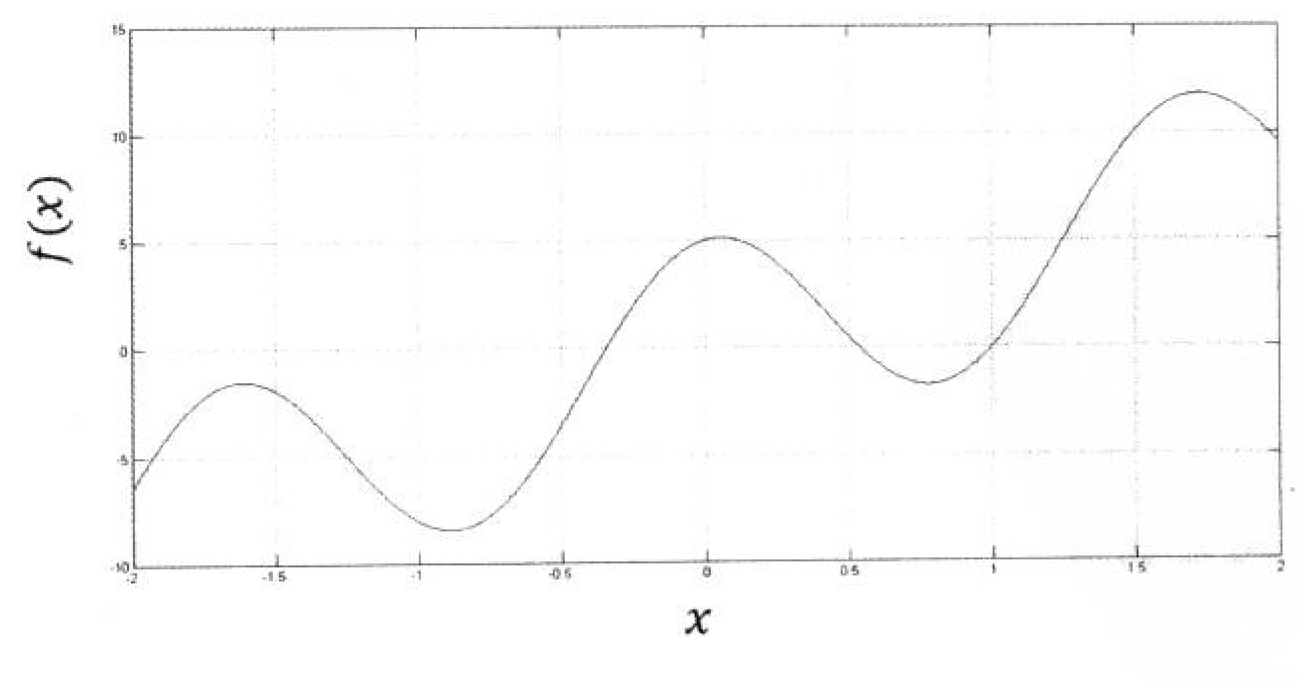


图 10-3

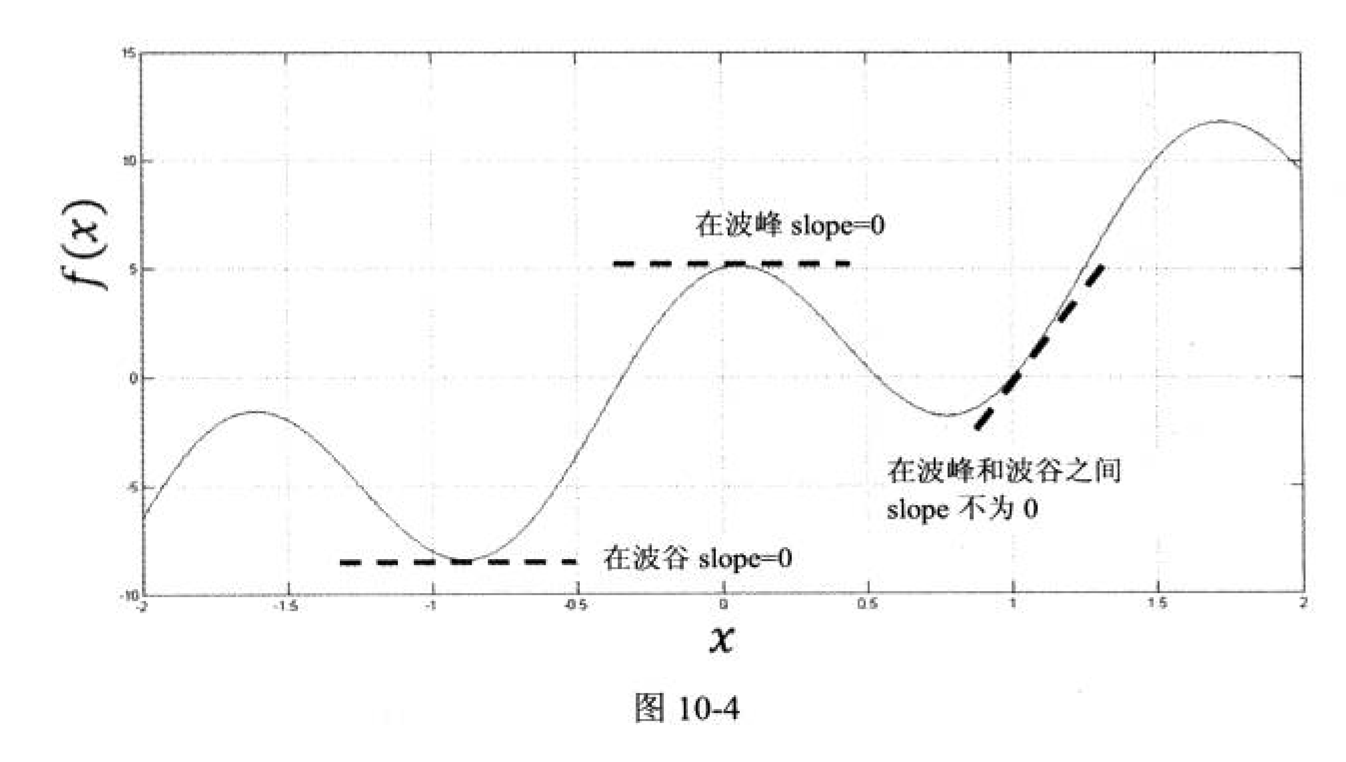


图 10-4

求解这一类问题，要根据曲线的斜率变为 0 这个特征，确定最大值和最小值的位置。详细的求解过程在微积分的课程里有介绍，在我们的讨论中，只需要知道这样一个道理：利用微积分可以求出斜率为 0 的点，并且可以确定这些点的 x 值。

例如，下面是一个抛物线函数，我们要求它的最小值：

$$\text{最小值} = 2x^2 + 8x - 6 \tag{10-11}$$

利用微积分的原理，对上述函数求导数，就可以得到该曲线的斜率。因此它的斜率方程是：

$$\text{斜率} = \frac{\mathrm{d}y}{\mathrm{d}x} = 4x + 8 \tag{10-12}$$

令斜率等于 0，得到 x=-2。这个结果表明，当 x=-2 时，这个抛物线函数取最大值或最小值。

仔细分析图 10-5 所示的曲线，我们发现该函数确实是在 x=-2 到达最小值。此外，必须提醒读者，该函数没有最大值，因为当 x 趋向无穷大时，y 值也趋向无穷大。

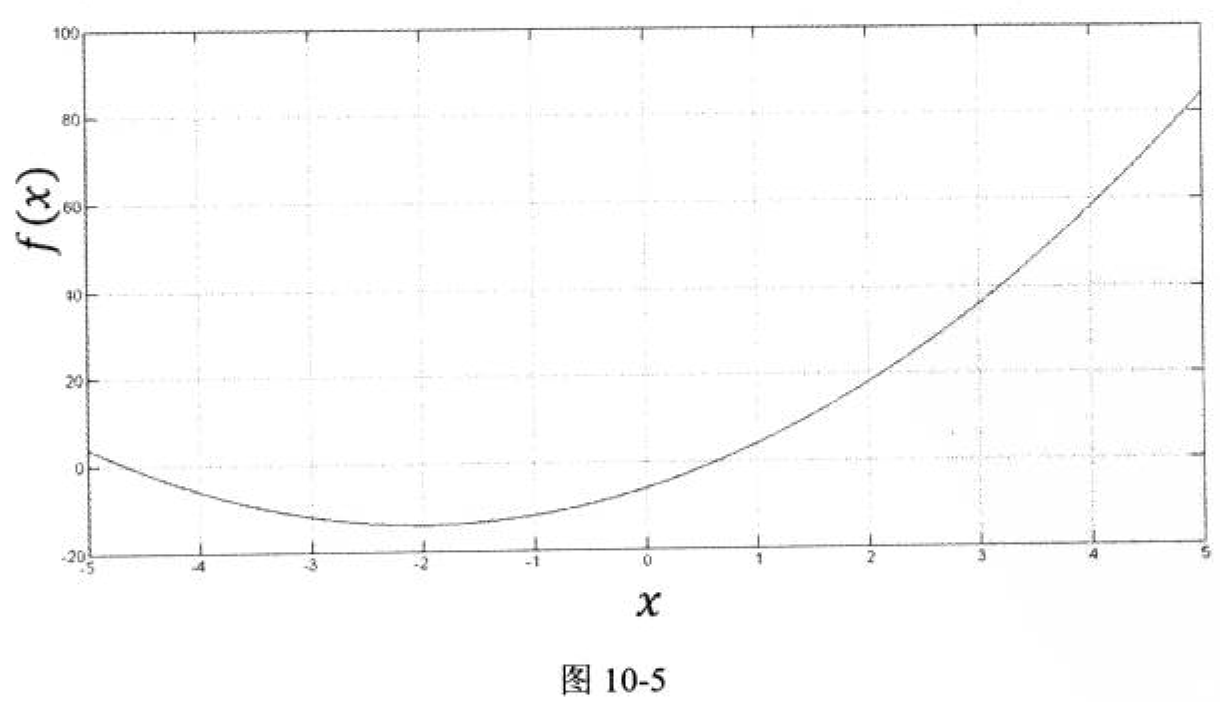


图 10-5

需要说明的是，在本例中，先确定了函数在 x=-2 位置的斜率为 0，然后再利用图形确定该点是最大值还是最小值。虽然根据微积分的原理，用二阶导数值可以确定该点是最大值还是最小值，这在前面第 9 章里曾提到过，但是这里我们把重点放在图形解法。

为了深入理解这个方法，我们分析与下面的目标函数有关的单变量最优化问题：

$$最小值 = x^3 - 2x^2 + x + 1 \tag{10-13}$$

该函数的斜率是它的导数：

$$斜率 = \frac{dy}{dx} = 3x^2 - 4x + 1 \tag{10-14}$$

令斜率等于 0，求得它的两个根为 x=1/3 和 x=1。这两个点是最大值还最小值？为此，我们需要分析函数的曲线。为了方便，我们只画出该函数在 - 0.2≤x≤1.5 范围内的曲线，如图 10-6 所示。

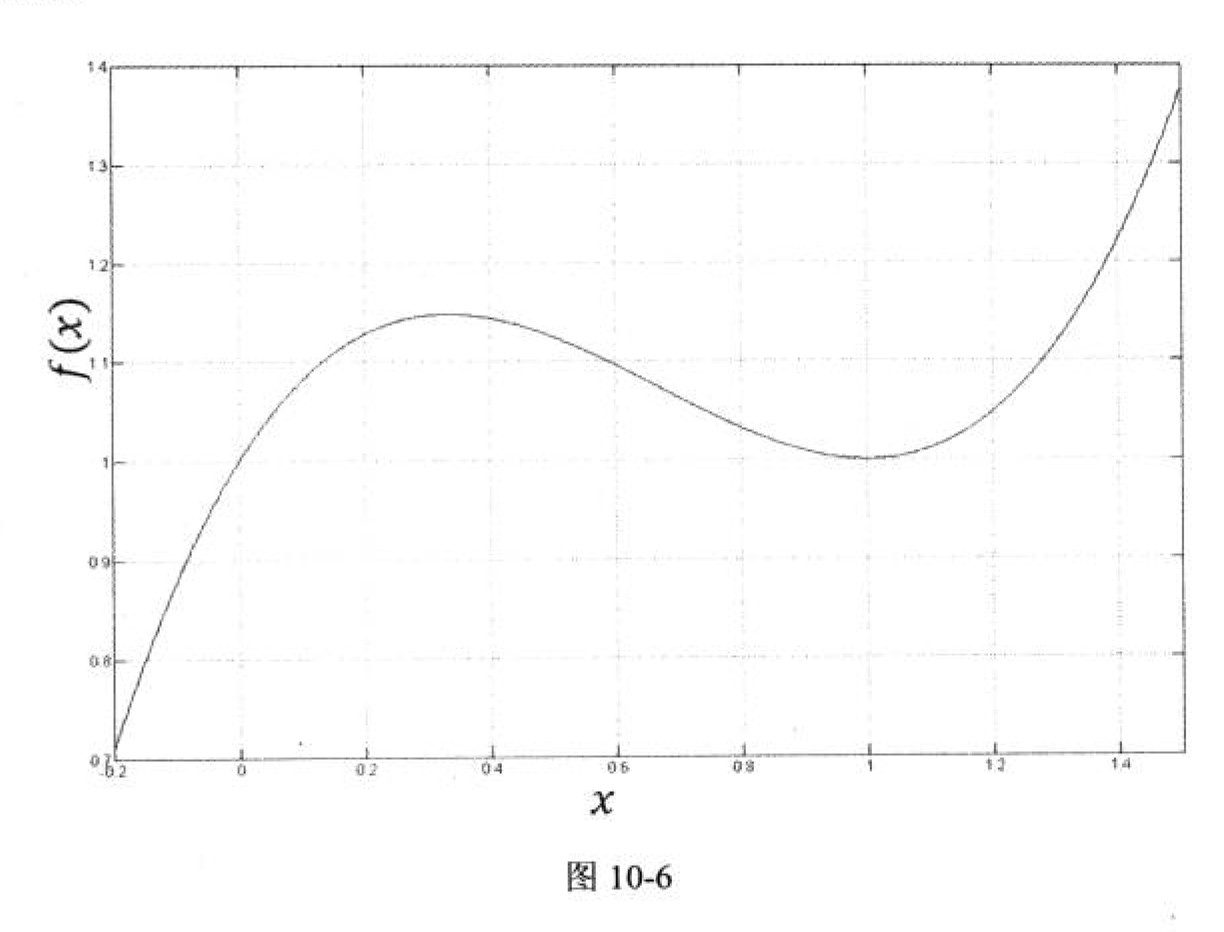


图 10-6

我们发现这两个点确实是极值点。x=1/3 对应于曲线的波峰，x=1 对应曲线的波谷。但是，仔细分析这个函数在整个范围内的曲线，发现 y 没有确定的极值，因为当 x 趋向无穷大时，y 值越来越大，同样，它也没有确定的最小值，因为当 x 趋向负无穷大时，y 值越来越小。

这些结果引入两个重要的概念，即局部最大值和局部最小值。x=1 点只能说是某种意义上的“最小值”，因为它只是在一个小范围内的最小值。如果我们把图 10-6 的曲线看成地图，我们是徒步旅行者，则处在 x=1 位置，围观四周，发现确实处于这次远步的最低点位置。由于不可能回过头看到整个地形的全貌，因此我们并不知道，其实离我们很远的地方，还有比这更低的位置！利用这个比喻，我们称 x=1 为局部最小值，称真正的最小值为全局最小值。这样的说法对 x=1/3 同样适用，它是局部最大值。

我们再举一个例子说明非线性无约束最优化的另一个重要性质：

$$最小值 = \sin(x) \tag{10-15}$$

它的斜率公式是

$$斜率 = \frac{\mathrm{d}y}{\mathrm{d}x} = \cos(x) \tag{10-16}$$

根据上述斜率公式，我们得到一个非常有趣的结果。有无数个点的斜率为 0。

$$x = \frac{\pi}{2}, \frac{3\pi}{2}, \frac{5\pi}{2}, \frac{7\pi}{2}, \frac{9\pi}{2}, \cdots \tag{10-17}$$

观察图 10-7 的曲线，证实了这个结论。

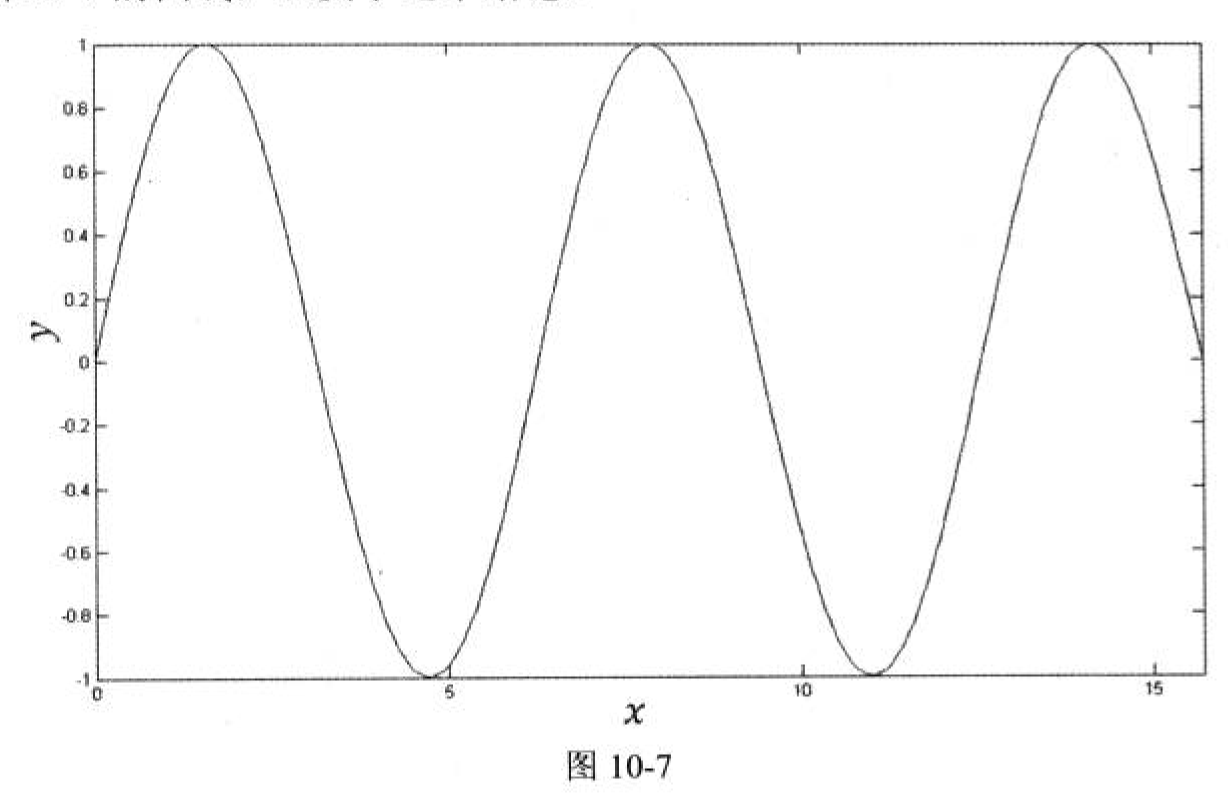


图 10-7

这个曲线图表明，有很多个全局最大值出现在 x=π/2,5π/2,9π/2,⋯位置，同时，也有很多个全局最小值出现在 x=3π/2,7π/2,11π/2,⋯位置。

回过头看本节的 3 个例子，我们得到关于非线性无约束最优化问题的几个结论：

- 斜率为 0 的点表示它们是极值点(极大值/极小值)。
- 斜率为 0 的点可能是局部极值点，也可能是全局极极点。在分析极值点时，不能掉以轻心。对于全局极极值点必须加以验证。
- 全局极值点可能并不唯一，特别是当目标函数为周期函数时，尤其如此。

虽然，本章的例子只限于单变量的目标函数，但是结论可以推广到多变量目标函数。对于多变量最优化问题，虽然用手工方法求函数的斜率不如单变量容易，但是本节介绍的基本原理有助于我们判断和利用数值解法的结果。

10.3.2 线性约束最优化问题

在第 10.2 节里，我们把由线性目标函数和线性约束方程组成的最优化问题归类为线性约束最优化问题。本节通过 4 个例子，详细介绍这类问题的求解方法。

分析下面这个问题，它有两个设计变量，三个约束方程。

最大值:

$$y = 3x_1 - 2x_2$$

约束条件：

$$x_1 \geq 3$$
$$x_2 \geq 1$$
$$2x_1 + x_2 \leq 10 \tag{10-18}$$

前面曾说过，约束方程给设计变量的取值加上限制。我们把这个问题的约束条件绘制成图 10-8，从这个图中可以看到设计变量允许的取值范围。

图 10-8 表明，设计变量的取值范围是一个由 3 个约束方程组成的封闭区域。我们称这个区域为可行解区域。在本例中，这个可行解区域是一个封闭多边形，而且多边形的边界就是约束方程。我们称它为封闭可行解区域，或带边界封闭区域。虽然从数学上讲并非所有的最优化问题都有封闭可行解区域，但是很多工程问题的可行解区域都是封闭的，这是因为存在物理约束条件限制设计变量的取值。当工程问题的一组约束条件组成一个封闭区域时，我们将得到一个令人感兴趣的结果。

为了说明结果的性质，我们根据设计变量的各种组合，画出目标函数。线性目标函数呈现出等值线的形状，如图 10-9 所示。

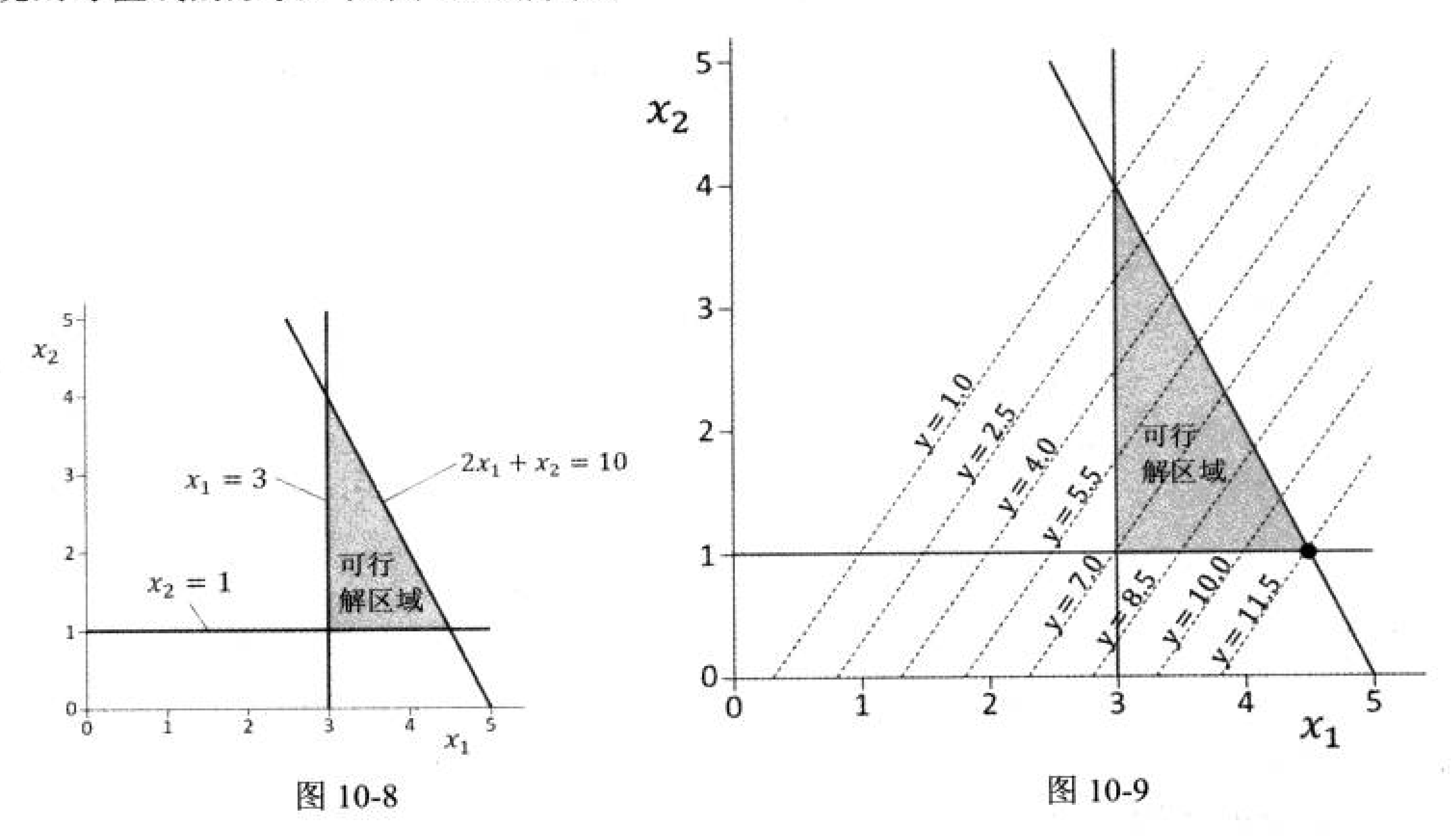


图 10-8　　　　图 10-9

在每条轮廓线上，目标函数的值都相等。目录函数的最小值出现在解区域的左上角(x_1=3，x_2=4，y=1)。等值线相互之间平行，而且随等值线从左上角到右下角，目标函数的值逐渐增加，直到与可行解区域相交为止。目标函数的最大值出现在可行解区域的右下角位置，即 x_1=4.5,x_2=1，y=11.5。这正是这个最优化问题的解。

虽然对于不同的目标函数和不同的约束条件，可行解区域的形状各不相同，等值线的斜率也各不一样，但是这个例子说明的解的这种特性具有普遍意义。由于线性目标函数和线性约束方程本身的特性，这类问题的极值点(最大值或最小值)出现在可行解区域的角点位置。在后面将要介绍的一些复杂问题里，不管设计变量有多少，也不管约束条件有多少，用到的算法解总是判断可行解区域的角点位置。一种名为“单纯形算法”的标准算法经常用来解这一类问题。

为了分析线性约束最优化问题的解的性质。我们分析下面这个例子。这个例子的约束条件与前一个例子相同，但是目标函数不一样。

最大值:

$$y = 4x_1 + 2x_2$$

约束条件:

$$\begin{aligned} x_1 &\geq 3 \\ x_2 &\geq 1 \\ 2x_1 + x_2 &\leq 10 \end{aligned} \tag{10-19}$$

图 10-10 画出了这个问题的可行解区域和目标函数的等值线。

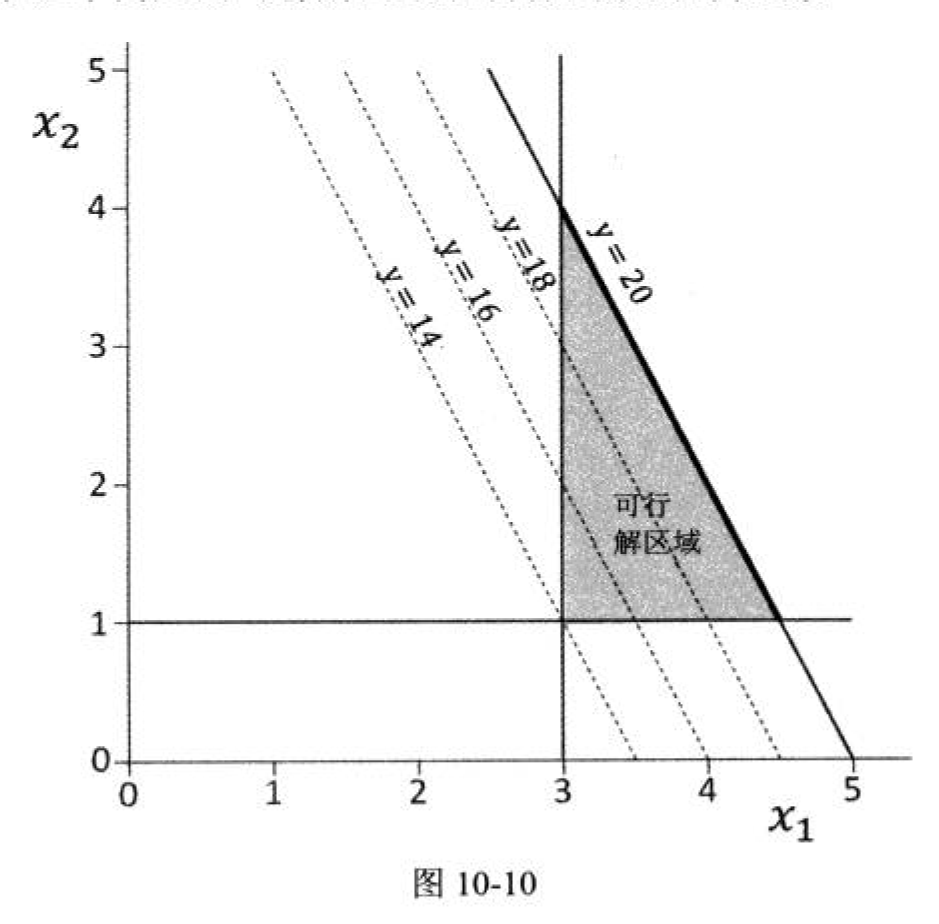


图 10-10

极小值出现在可行解区域的左下角位置，即 x_1=3，x_2=1，y=14。注意，目标函数的斜率与可行区域右上角边界约束条件的斜率相等。因此目标函数的最大值出现在可行区域的右上方边界线上，即方程为 $4x_1+2x_2=20$ 的直线上。它包括两个角点(分别为 x_1=4.5，x_2=1 和 x_1=3，x_2=4)和边界线上这两个角点之间的任意点。这也表明，虽然目标函数的极值点确实出现在角点位置，但是这个线性约束问题的解却不是唯一的，除了两个角点外，目标函数在两个角点之间任意一点上都有最优解。当我们以后面对越来越复杂的最优化问题，并且需要例用计算机的算法求解法求解问题时，必须牢记一点：我们求得的结果可能不是问题的唯一解，它可能是众多最优解中的一个。

正如本节前面曾提到，不是所有的线性约束最优化问题都存在一个封闭的可行区域。在有些情况下，我们还必须考虑解的另一种可能性。例如，我们对前面这个例子稍加修改，去掉其中一个约束条件，即得到如下的最优化问题。

最大值:

$$y = 4x_1 + 2x_2$$

约束条件:

$$x_1 \geq 3$$
$$x_2 \geq 1 \tag{10-20}$$

与这一组约束条件对应的可行解区域如图 10-11 所示。

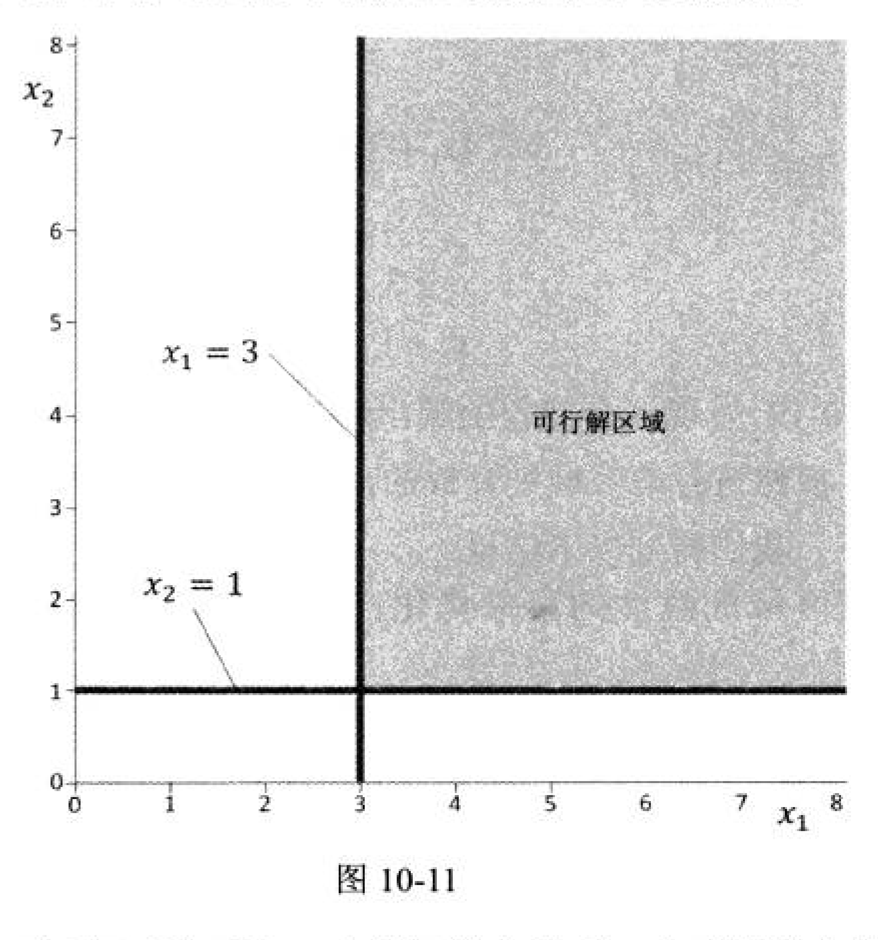


图 10-11

在这个例子里，可行解区域不是一个封闭的多边形。由于设计变量没有上限约束条件，因此这个可行解区域在两个方向都可以扩展到无穷区域。我们称这样的可行解区域为开放可行解区域或者无边界可行解区域。在这里增加目标函数的等值线，得到如图 10-12 所示的图形。我们可以得出结论：这个问题的最大值不受约束，因此最大值出现在无穷远处。

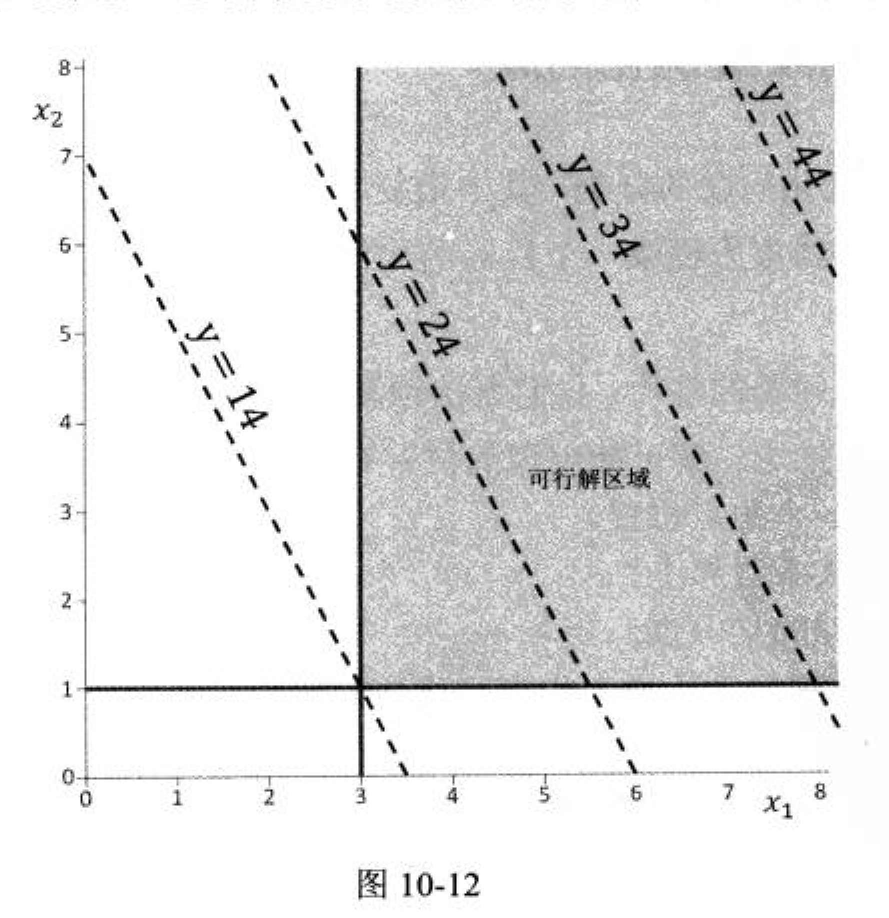


图 10-12

这个两变量例子说明了线性约束最优化问题解的另一个性质。当问题的可行解区域是开区域时，它的解可能不受约束。虽然解不受约束，但是也有可能出现在可行解区域的某个角点上。下面这个例子正好说明了这一点：

最小值：

$$y = 4x_1 + 2x_2$$

约束条件：

$$x_1 \geq 3$$
$$x_2 \geq 1 \tag{10-21}$$

我们再来分析图 10-12，发现这个问题的解出现在可行区域的角点上(x_1=3，x_2=1，y=14)，因此，我们得出结论：虽然可行解区域是开放的，但是最优解可能出现在某个角点位置。

根据前面这些例子，我们得到以下结论：

- 带封闭可行解区域的线性约束最优化问题的解总是出现在可行解区域的角点位置。
- 带开放可行解区域的线性约束最优化问题的解或者出现在可行解区域的一个角点上，或者不受约束。
- 线性约束最优化问题的解不是唯一的，可能不止一个角点出现符合最优化解的条件。

这些结论是针对两变量的，但是可以推广到任意个变量和任意个约束条件。随着我们要解决的问题越来越复杂，求解这些复杂的问题不能借助图形工具，就要用这些结论进行算法求解，并解释求解结果。

10.3.3　非线性约束最优化问题

第 10.2 节给出最优化问题的分类，其中最复杂的一类是非线性约束最优化问题。前面曾说过，如果其中一个约束方程或一个目标函数包含非线性五项原则，则这个问题就是非线性优化问题。

根据非线性最优化问题的性质，我们无法预测解的位置。它肯定出现在可行解区域内，但是无法得出其他几个推论。解可能出现的位置有：

- 可行解区域的一个角点位置。
- 可行解区域的某条边界上。
- 可行解区域内部的某一点。

此外，在第 10.3 节里的非线性无约束最优化问题的几个推论仍然有效：

- 求解方法得到的结果可能是局部极值点而非全局极值点。
- 解可能不唯一(即，目标函数可能在多个点得到同一个最优值)。

由于非线性约束最优化问题存在不可预测性，因此经常需要用复杂的算法求解这类问

题，而且得到的结果在应用之前必须经过认真的验证和分析。

10.4 用 MATLAB 求解最优化问题

MATLAB 标准函数工具箱(funfun 工具箱)提供了两个求解最优化问题的函数：

- fminsearch( )，用来求解非线性无约束多变量最优化问题。
- fminbnd( )，只能用来求解非线性约束最优化问题中特定的一类，即只含有一个设计变量的问题。

大多数标准版 MATLAB 都带有这两个函数。要提醒读者，MATLAB 还有最优化工具箱，它提供了许多功能强大的工具，用来求解很多类型的最优化问题。标准版 MATLAB 没有这个工具箱。

10.4.1 教程：用 fminsearch( )求解非线性无约束最优化问题

例 10.1

单变量函数的最小值问题。

求使下面这个函数最小化的 x 值：

$$y = 3x^4 + x^3 \cos \pi x \tag{10-22}$$

解：

首先，我们建立一个 MATLAB 函数，它可以求目标函数的值。启动 MATLAB，新建一个 m-file，输入以下命令：

```
1 function y=objfunc(x);
2 y=3*x^4+x^3*cos(pi*x);
```

把这个脚本程序保存为 objfun 文件。

现在我们用 fminsearch( )函数找出使这个目标函数最小化的 x 值。算法要求输入一个估测值为 x 的初始搜索值。由于这个目标函数只有一个变量，因此我们利用图形确定它的初始值。在 MATLAB 的命令窗口里，输入以下命令：

```
>> fplot('objfunc',[-1 1])
```

出现如图 10-13 所示的曲线。

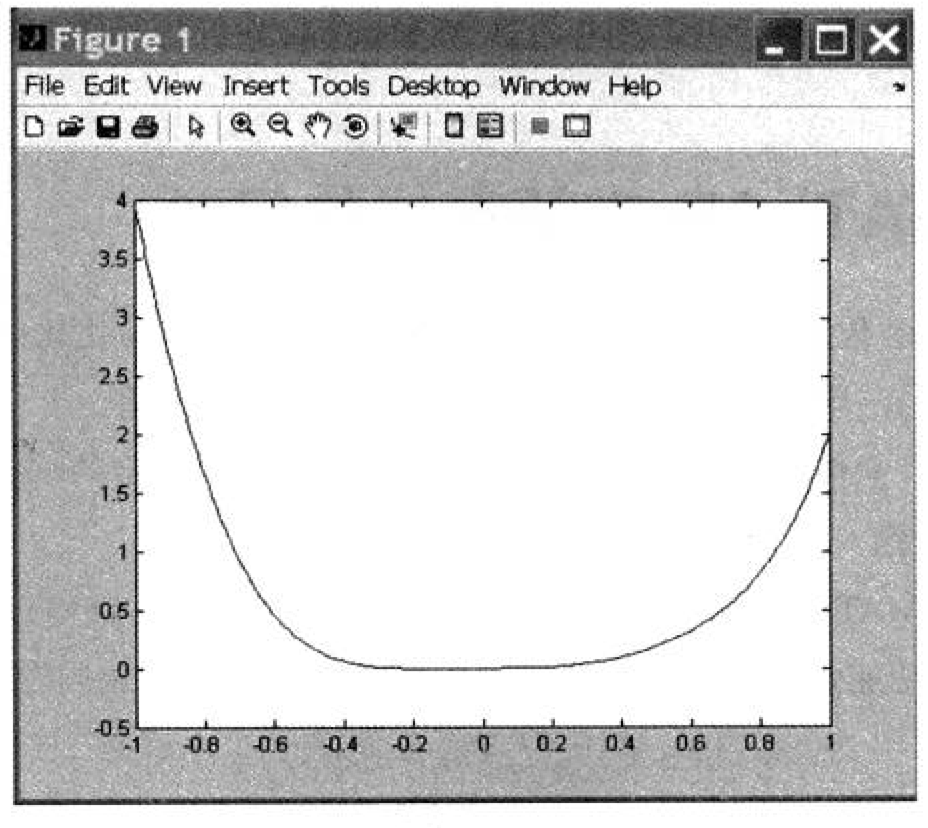


图 10-13

从这个图形可以看出，最小值出现在 $x=-0.2$ 附近。我们就把它作为初始值，输入以下命令：

```
>>xmin=fminsearch('objfunc',-.2)
```

这个命令指示 MATLAB 在 $x=-0.2$ 附近搜索 x 的值，使得定义在 objfunc.m 文件里的目标函数值最小，并把结果保存在 *xmin* 变量里。这个命令执行后，出现如下的结果：

```
xmin =
        -0.1834
```

如果我们还想知道目标函数在这个位置的值，只需要把这个值代入 objfunc( )函数。输入如下命令：

```
>> ymin=objfunc(xmin)
ymin =
        -0.0018
```

以上结果表明，当 $x=-0.1834$ 时目标函数达到最小值 $y=-0.0018$。

运算结果表明，这是一个“良态”(well-conditioned)问题，即运算过程中没有出现“陷阱”，这是最优化算法解中经常出现的。前面的曲线也表明，这个问题只有一个最小值。为此，我们用不同的初始值，再次调用此命令：

```
>>xmin=fminsearch('objfunc',100)
```

不论们用什么样的初始值进行测试，得到的结果肯定都差不多。但是对于其他问题，情况并非总是如此，在下一个例子里读者就会发现。

例 10.2

单变量函数的局部最小值问题。

求 x 的值，使下面这个函数最小化：

$$y = 3x^4 + x^3 \cos 10\pi x \tag{10-23}$$

解：

我们先建立一个 MATLAB 函数用来计算这个目标函数的值。打开 objfunc 文件，修改第二行的内容，如下：

```
1. function y=objfunc(x);
2. y=3*x^4+x^3*cos(10*pi*x);
```

保存文件。

与前面一样，我们先画出这个函数的曲线，决定初始值。

```
>> fplot('objfunc',[-1 1])
```

图 10-14 就是这个目标函数的曲线。

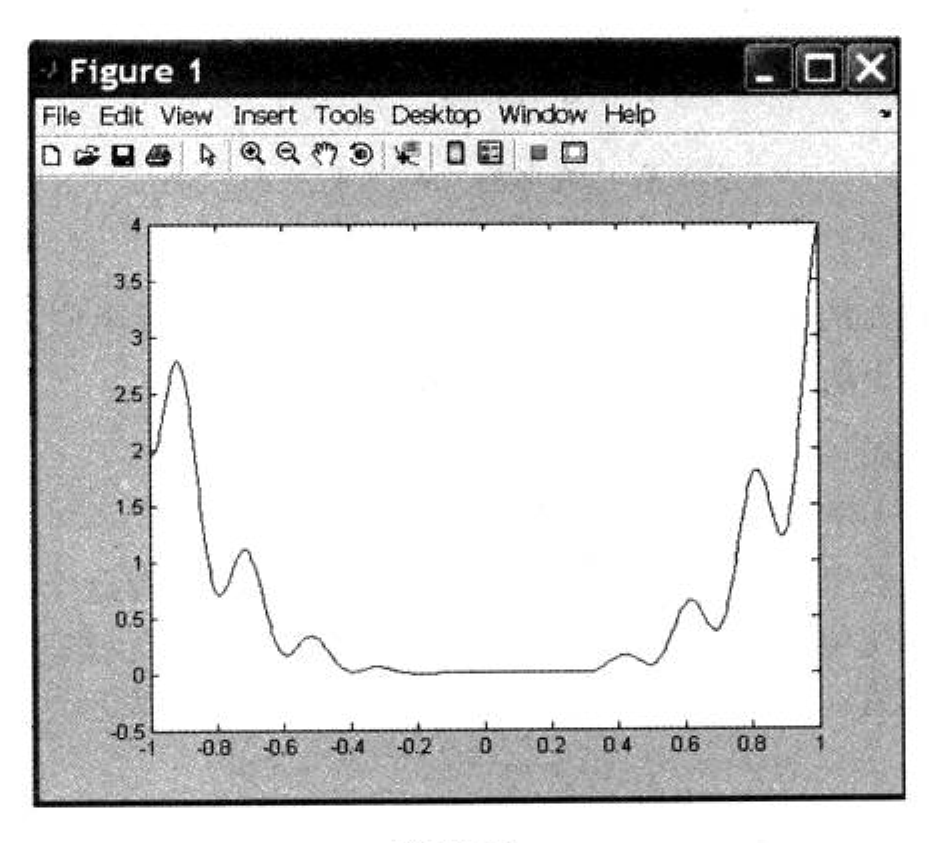


图 10-14

现在的曲线明显不同于图 10-13，因为它有许多局部最小值。开始我们也在 $x=-0.2$ 邻域内搜索。执行 fminsearch 命令，初始值设为 -0.2：

```
>>xmin=fminsearch('objfunc',-.2)

xmin =
    -0.2028
```

计算目标函数在这个位置的值。

```
>> ymin=objfunc(xmin)

ymin =
```

```
        -0.0032
```

结果表明，当 x 等于 - 0.2028 时，目标函数达到最小值 - 0.0032。

现在我们尝试用其他的初始值。例如，取初始值为 - 0.6，重新执行 fminsearch 命令：

```
>> xmin=fminsearch('objfunc',-.6)

xmin =
      -0.5928
```

这个结果明显不同于前面的结果。回过头来分析图 10-14，我们发现在 $x=-0.6$ 附近也出现了一个局部最小值。因此，本例的算法解得到的是一个局部最小值，而不是全局最小值。这个函数只有一个变量，因此很容易作出它的曲线图，并且很容易得到结果，我们得到的结果并非真正意义的“最优值”。但是对于多变量问题，用图形来验证通常是行不通的。

例 10.3

多变量函数的最小化问题。

求 x_1、x_2 和 x_3 的值，使下面的函数最小化：

$$y = x_1^2 - x_1 + x_2^4 - x_2 + x_3^4 + x_3^2 - x_3 \tag{10-24}$$

解：

我们从前面刚建立的 MATLAB 函数开始，求这个新的目标函值。打开 objfunc 文件，修改第二行内容，如下所示：

```
1 function y=objfunc(x);
2 y=x(1)^2-x(1)+x(2)^4-x(2)+x(3)^4+x(3)^2-x(3);
```

注意，在这个函数里，变量 x 是一个三个元素的数组。保存这个函数。

由于这是一个多变量函数，因此无法绘制它的图形，也就无法确定变量的初始值。只好任意设置一个初始值。在本例，我们取 $x_1=x_2=x_3=0$。用数组表示这个初始值，输入以下命令：

```
>> xmin=fminsearch('objfunc',[0 0 0])

xmin =
      0.5000        0.6300        0.3855
```

现在求目标函数在该点的值：

```
>> ymin=objfunc(xmin)

ymin =
      -0.9373
```

这个结果表明，当 $x_1=0.50$，$x_2=0.63$，$x_3=0.39$ 时，这个目标函数达到最小值 - 0.94。

它是全局最小值吗？实际上，无法确定它是否是一个真正的最小值。为了测试它是否是真正的最小值，我们用不同的初始值进行测试，例如，我们执行以下命令：

```
>> xmin=fminsearch('objfunc',[-10 -10 -10])

xmin =
     0.5000  0.6299   0.3854
```

我们再用其他初始值进行测试：

```
>> xmin=fminsearch('objfunc',[50 -50 50])

xmin =
       0.5000    0.6300    0.3855
```

上述测试表明，用相差很大的初始值进行测试，得到的结果保持不变，因此完全有理由相信这是一个全局最小值。

10.4.2 教程：用 fminbnd( )求解非线性约束最优化问题

fminbnd( )函数可以用来求解非线性约束最优化问题，但是要受到以下限制：

- 只能求解单变量问题。
- 约束条件只能用来设置设计变量 x 的取值极限(即 $x_{\min} \le x \le x_{\max}$)。

例 10.4

求单变量函数的最小值。

求 x 的值，使下面这个函数最小化：

$$y = 3x^4 + x^3 \cos 10\pi x \tag{10-25}$$

x 的取值范围为 $0.8 \leqslant x \leqslant 1.0$。

解：

我们发现这个目标函数与前面例 10.2 的目标函数完全相同。先建立一个 MATLAB 函数，计算这个目标函数的值，打开 objfunc( )函数，修改第二行的内容，如下所示：

```
1 function y=objfunc(x);
2 y=3*x^4+x^3*cos(10*pi*x);
```

保存程序。

fminbnd( )函数需要 3 个参数:目标函数名、设计变量的下限值和设计变量的上限值。输入并执行以下命令：

```
>> xmin=fminbnd('objfunc',.8,1)

xmin =
     0.8910
```

计算目标函数在这点的值：

```
>> ymin=objfunc(xmin)
```

```
ymin =
     1.2115
```

结果表明，目标函数在 x=0.89 时达到最小值 1.21。参考图 10-14 该函数的曲线，在给定的范围内这个结果是正确的。

10.5　用 Excel 求解最优化问题

Excel 的规划求解添加件是一个功能强大的工具，它可以求解包括非线性约束问题在内的各类最优化问题。如果读者的 Excel 已安装载了规划求解工具，它就会在数据选项卡里出现，如图 10-15 所示。



图 10-15

如果读者的 Excel 还没有安装规划求解添加件，则参考第 8.4 节的内容进行安装。

10.5.1　教程：用 Excel 求解无约束最优化问题

例 10.5

单变量无约束的最小化问题。

求 x 的值，使下面的目标函数最小化：

$$y = 3x^4 + x^3 \cos \pi x \tag{10-26}$$

解：

这个问题与例 10.1 完全相同。在例 10.1 里我们利用 MATLAB 的 fminsearch 命令。打开 Excel 电子表格，把电子表格设置成如图 10-16 所示。

在 B1 单元里输入最优化问题的初始估算值。为了与前面的例子保持一致，初始值取 0.2，输入初始值，如图 10-17 所示。

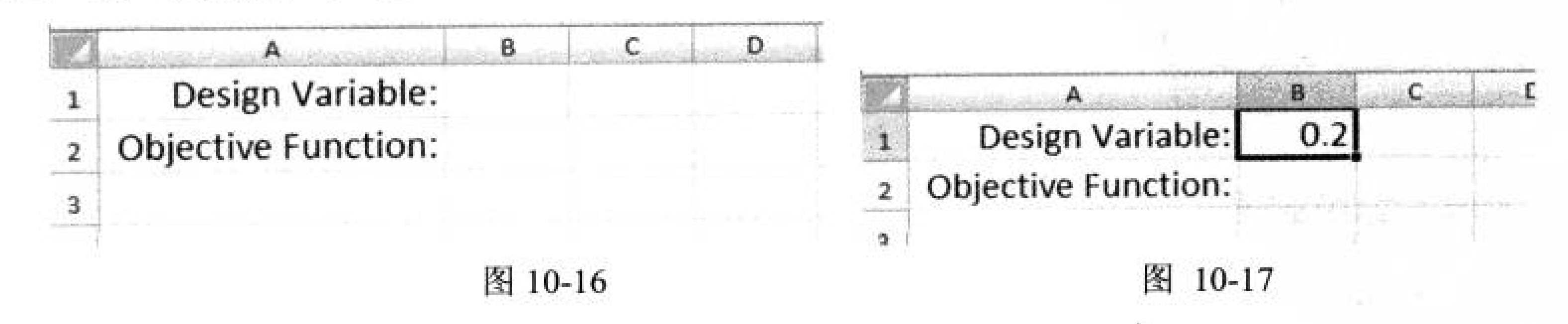


图 10-16　　　　图 10-17

在 B2 单元里输入表示目标函数的公式，公式要引用 B1 单元作为设计变量，如图 10-18 所示。

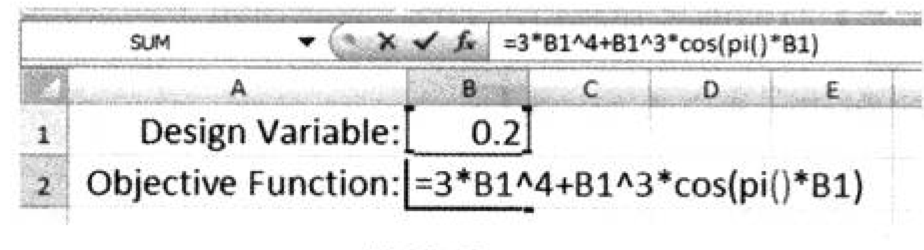


图 10-18

按 Enter 键，求目标函数在设计变量等于初始值时的值。

现在利用规划求解工具。在【数据】选项卡里，选择规划求解工具，如图 10-19 所示。



图 10-19

在弹出的【规划求解参数】对话框中，单击【设置目标单元格】右侧的选择按钮，如图 10-20 所示。

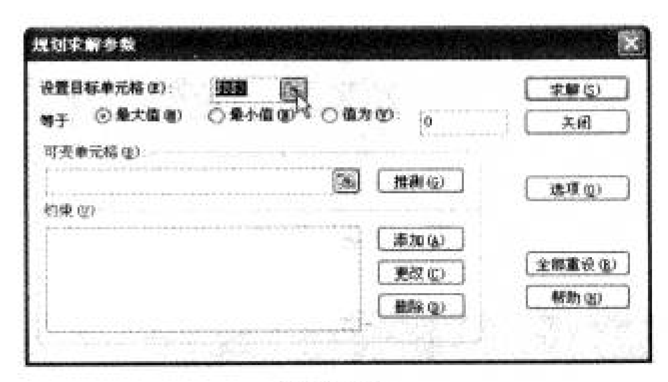


图 10-20

选择 B2 单元(这个单元里保存了目标函数)，如图 10-21 所示。

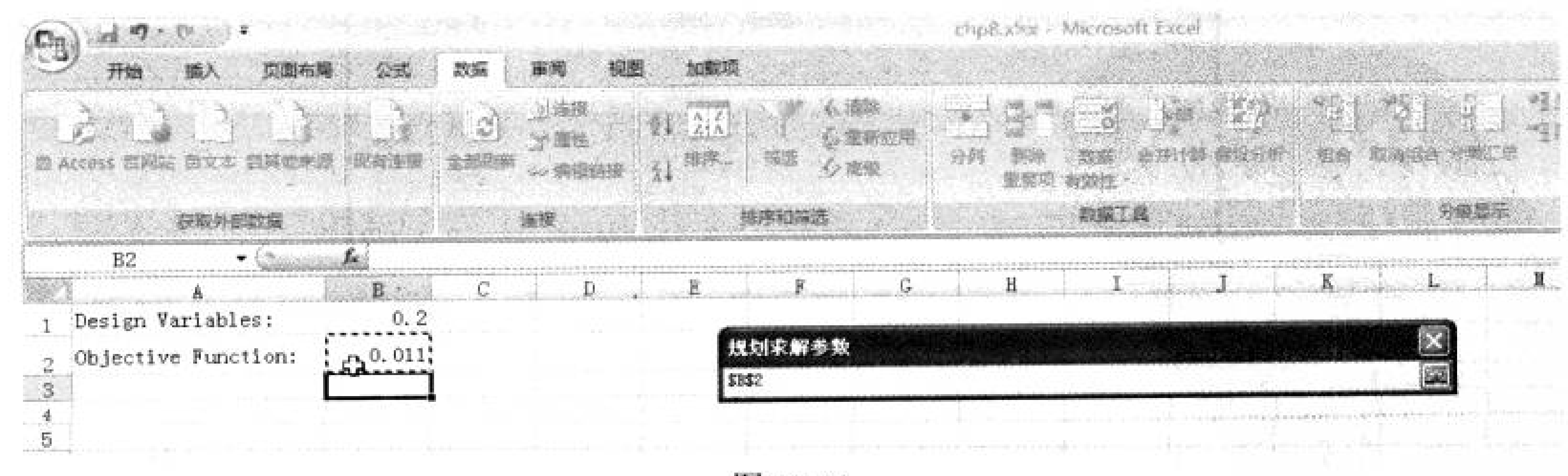


图 10-21

由于我们这个问题是一个最小值问题，因此在【等于】选项中选择【最小值】选项，如图 10-22 所示。

单击【可变单元格】右侧的选择按钮，如图 10-23 所示。

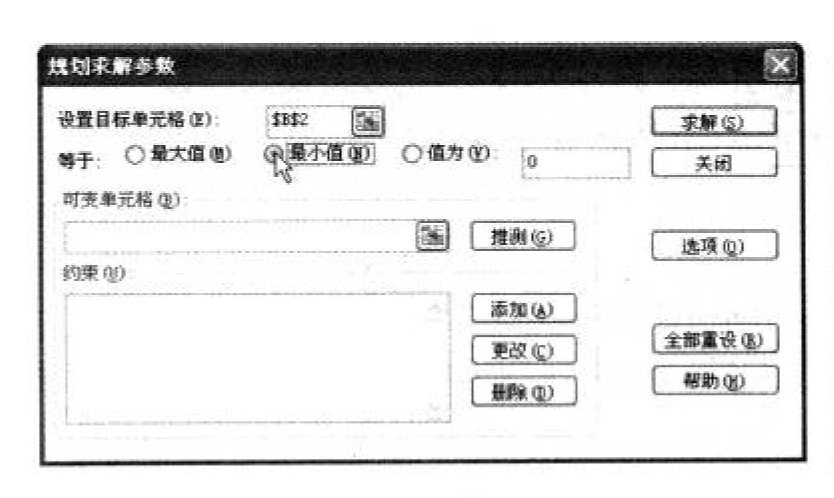


图 10-22

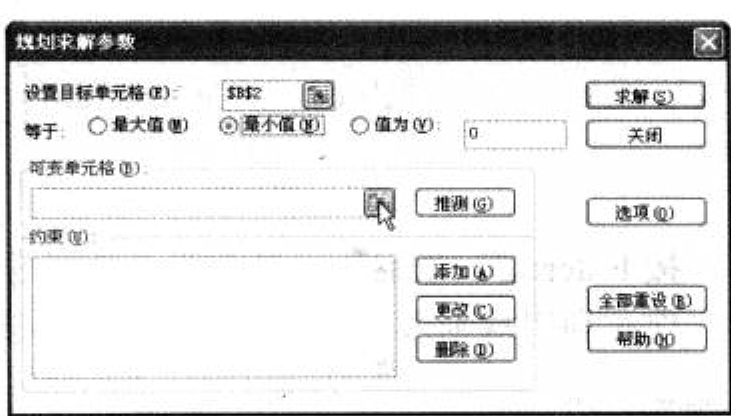


图 10-23

选择 B1 单元格(该单元格包含设计变量的初始值)，如图 10-24 所示。

由于这个问题没有约束条件，因此不需要设置约束条件，直接进行计算。为此，我们只需要单击【求解】按钮，如图 10-25 所示。

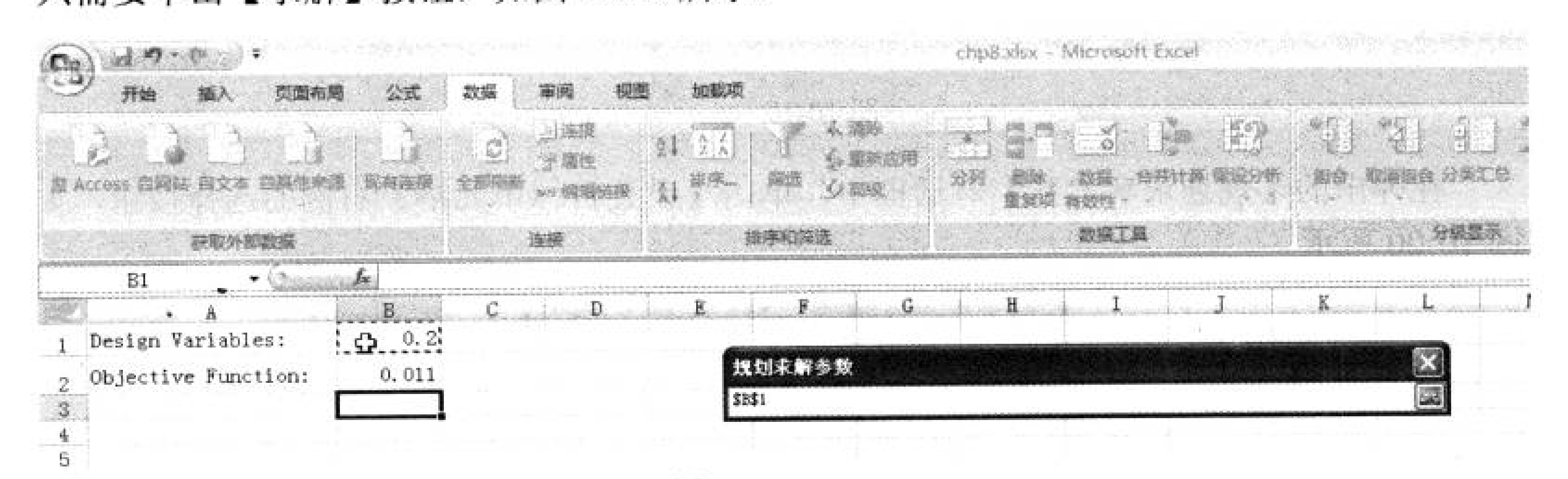


图 10-24

图 10-25

单击【确定】按钮，接受计算结果，如图 10-26 所示。

为了把这里的结果与前面的用 MATLAB 方法求得的结果进行比较，我们把 B1 和 B2 单元的小数位数设置为 4 位。结果如图 10-27 所示。

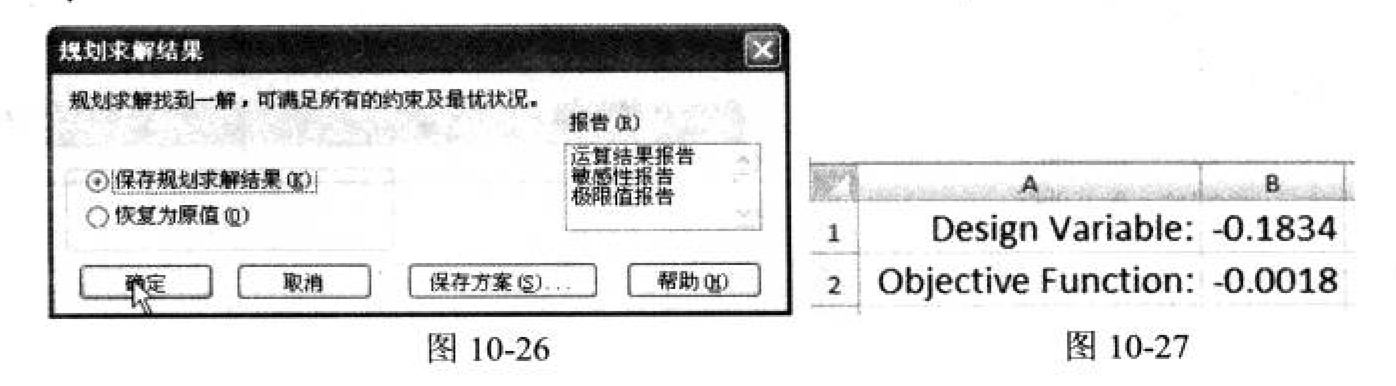


图 10-26　　　　图 10-27

这些结果与前面用 MATLAB 方法求得的结果完全相符。

例 10.6

多变量无约束最小化问题。

求 x_1、x_2 和 x_3 的值，使下面的函数最小化：

$$y = x_1^2 - x_1 + x_2^4 - x_2 + x_3^4 + x_3^2 - x_3 \tag{10-27}$$

解：

在 Excel 里新建一个电子表格。这里的 3 个变量的初始值与前面的 MATLAB 例子完全一样。把电子表格设置为图 10-28 的格式。

按 Enter 键，计算目标函数的初始值。

选择【数据】选项卡里的规划求解工具，把目标单元设置为 B6，把【等于】选项设置为【最小值】，由于这个问题是多变量最优化问题，因此在【可变单元格】里需要选择多个单元格。为此，单击它右侧的选择按钮，如图 10-29 所示。

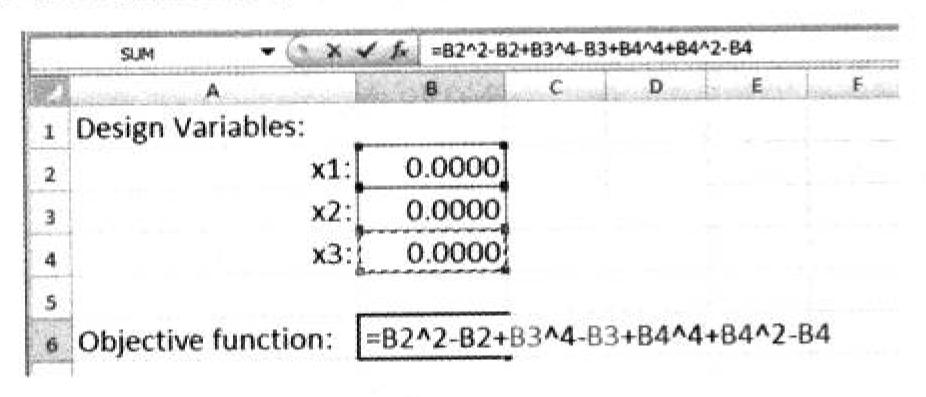


图 10-28

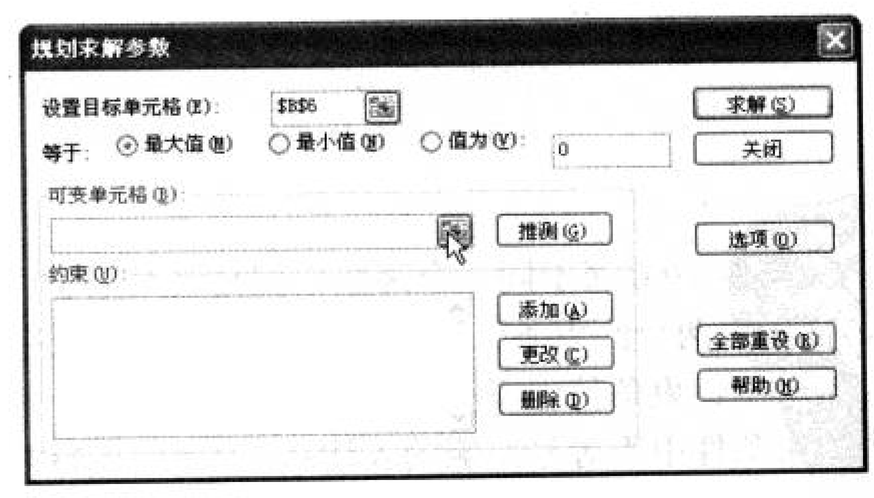


图 10-29

在 B2 单元里单击并拖动鼠标的左键，把它拖动到 B4，如图 10-30 所示。

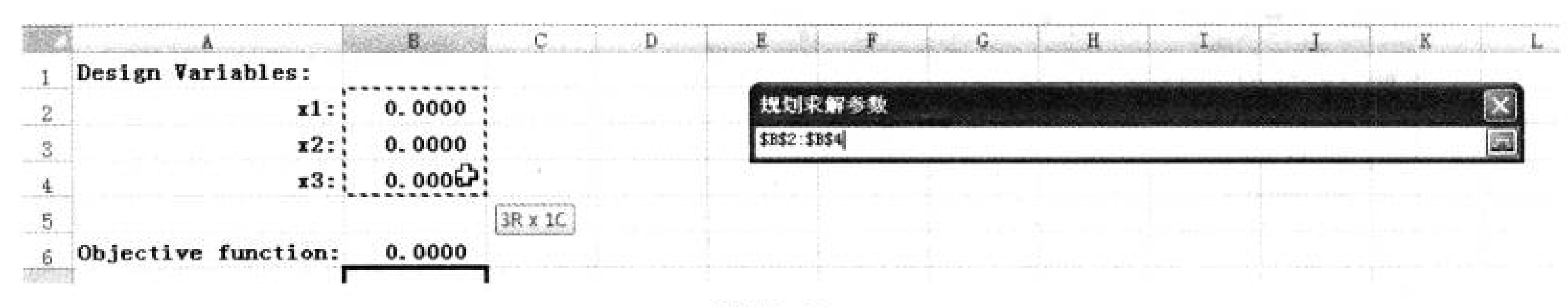


图 10-30

这会选择包含设计变量的全部单元。

单击规划求解工具，然后单击【确定】按钮，结果如图 10-31 所示。

	A	B
1	Design Variables:	
2	x1:	0.5000
3	x2:	0.6300
4	x3:	0.3855
5		
6	Objective function:	-0.9373

图 10-31

我们发现结果与 MATLAB 方法求得的结果完全相同。

10.5.2　教程：用 Excel 求解约束最优化问题

例 10.7

非线性约束函数的最小化。

求解下面的最优化问题。

最小值：

$$y = x_1^2 - x_1 + x_2^4 - x_2 + x_3^4 + x_3^2 - x_3$$

约束条件：

$$x_1 + 2x_2 \geq 4$$

$$x_1 - x_3 \leq 0.1 \tag{10-28}$$

解：

这个例子的目标函数与例 10.6 属于同一个函数，但是这个例子引入了两个约束条件。修改前面的电子表格，插入这两个约束条件。把例 10.6 里的电子表格修改成如图 10-32 的格式，在单元 B8 输入第一个约束条件中不等号左侧的表达式。

在 B9 输入第二个约束条件中不等号左侧的表达式，如图 10-33 所示。

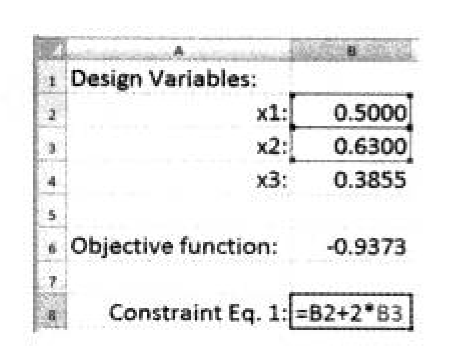


图 10-32

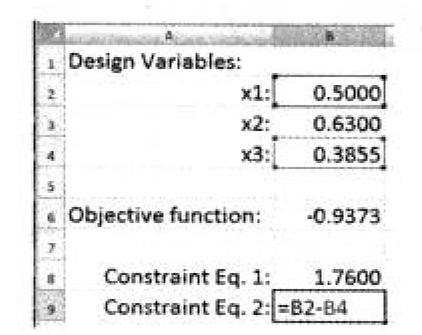


图 10-33

打开【数据】选项卡里的规划求解工具。规划对话框的设置与前面的例子一样(单元B6 是目标单元格，最小值，B2:B4 是可变单元)。如果不对，可重新设置这些值。现在要添加两个约束条件。单击【约束】条件右侧的【添加】按钮，如图 10-34 所示。

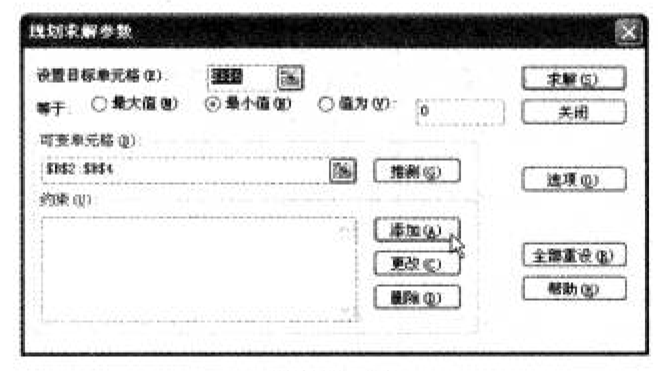


图 10-34

出现一个【添加约束】对话框，在【单元格引用位置】里选择第一个约束条件所在的单元(B8)，如图 10-35 所示。

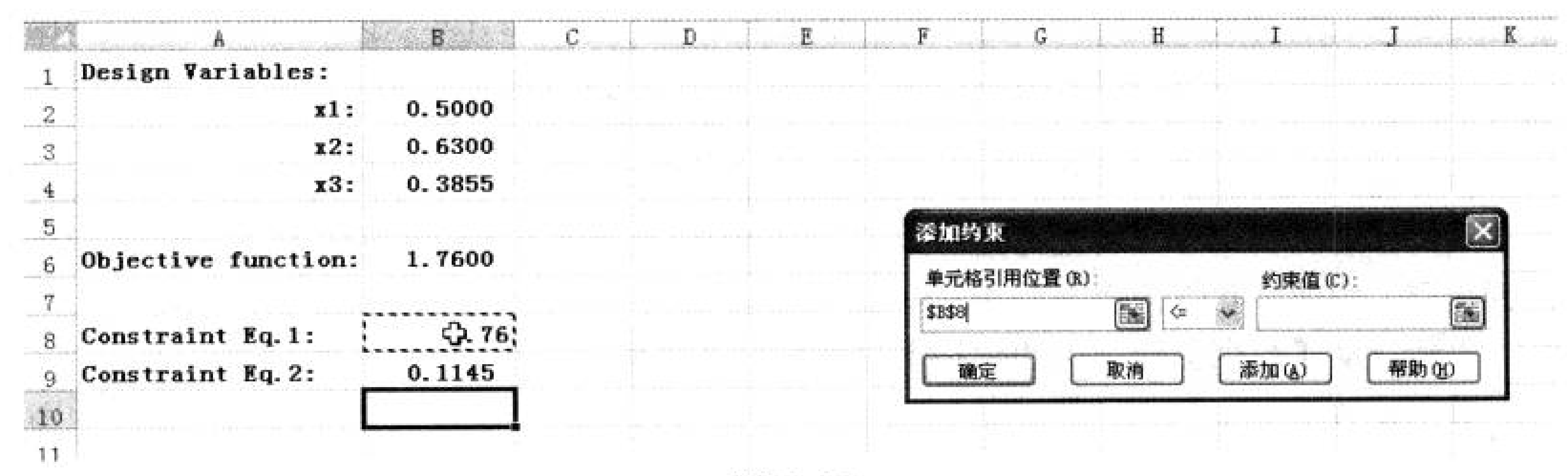


图 10-35

从下拉列表中选择大于等于号(>=)，如图 10-36 所示。

注意，这个下拉列表里除了等于和不等于号外，还可以把约束值设置为整数(int)，或 0 和 1(bin)。

在【约束值】输入框里输入约束方程右侧的值(这里是 4)，单击【确定】按钮，如图 10-37 所示。

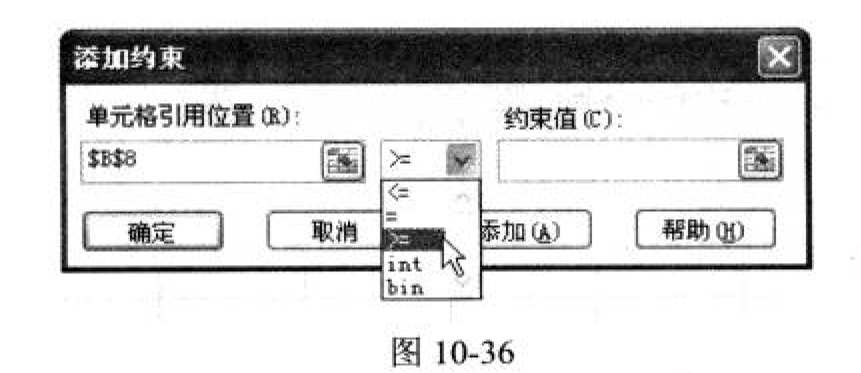


图 10-36



图 10-37

约束方程出现在【规划求解参数】对话框里，如图 10-38 所示。

单击【添加】按钮，重复前面的步骤，添加第二个约束方程。添加后，这个对话框如图 10-39 所示。

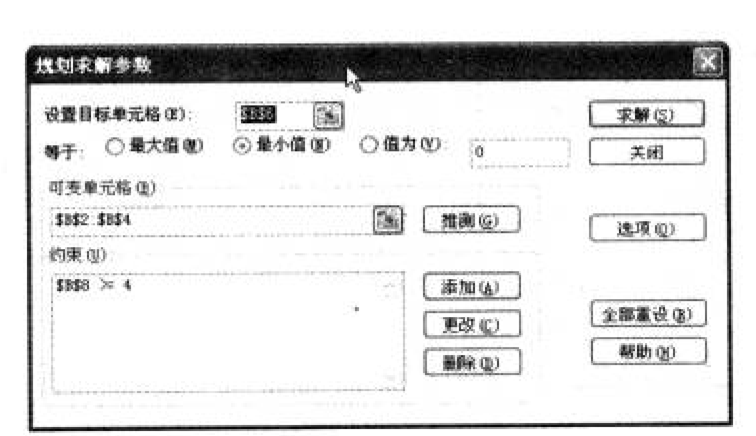


图 10-38

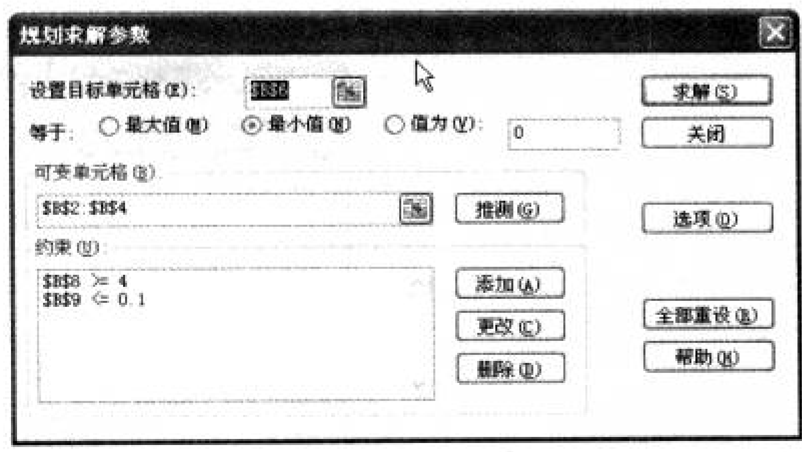


图 10-39

至此问题求解的全部参数都已设置完毕。注意，就像求解非约束问题一样，这里我们对设计变量不做任何修改。现在完全可以让规划求解工具开始计算。既然问题的参数都已确定，我们就可单击【求解】按钮，求解结束后，再单击【确定】按钮，接受计算结果，如图 10-40 所示。

正如前一节曾提到的，没有特别的办法验证这个结果是不是一个全局最优解，或者只是一个局部解。其中一个验证办法是用不同的初始值重新执行规划求解，看看得到的结果是否相同。为此，我们改变 B2、B3 和 B4 单元里的值，如图 10-41 所示。

重新启动【数据】选项卡的规划求解工具，前一次计算设置好的参数会被保留下来，只需要单击【求解】按钮，再次执行求解 ，单击【确定】按钮接受计算结果，结果如图 10-42 所示。

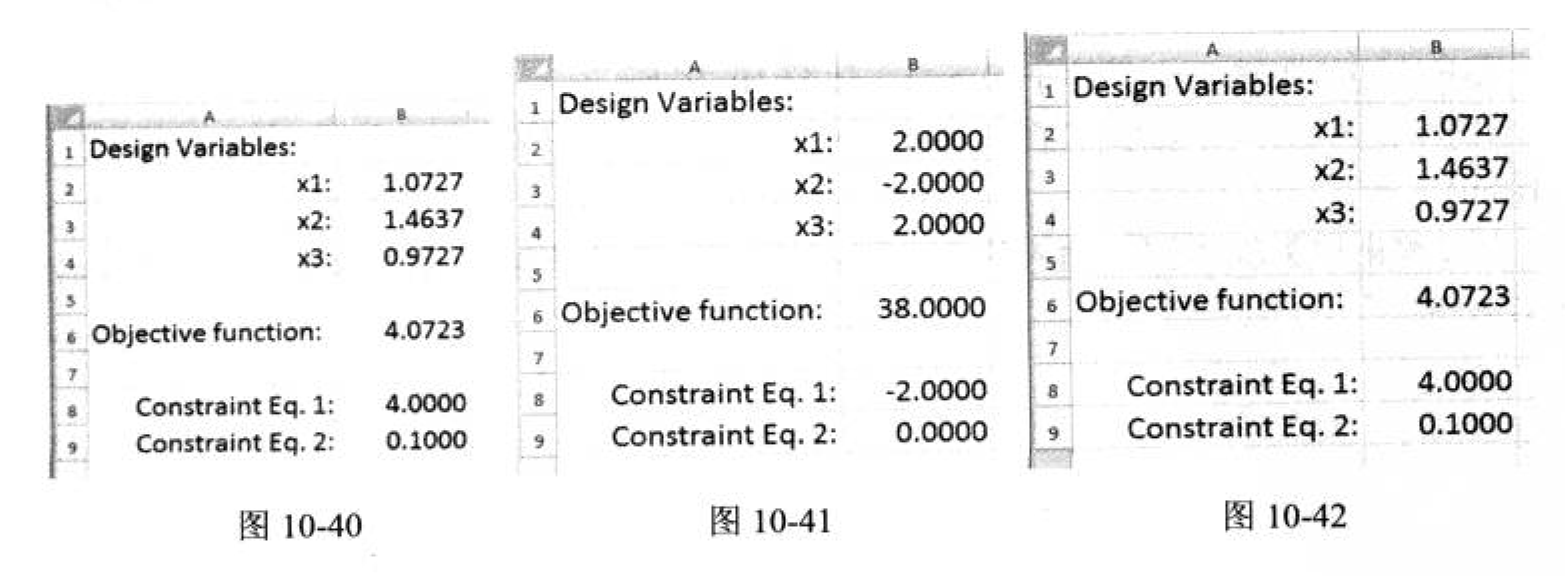


图 10-40　　图 10-41　　图 10-42

结果相同，虽然还是无法保证它是一个全局最优解，但是它确实让我们相信它是一个全局最优解。

例 10.8

线性约束函数的最小化问题。

求解下面的最优化问题。

最小值：

$$y = 3x_1 + x_2 + 5x_3$$

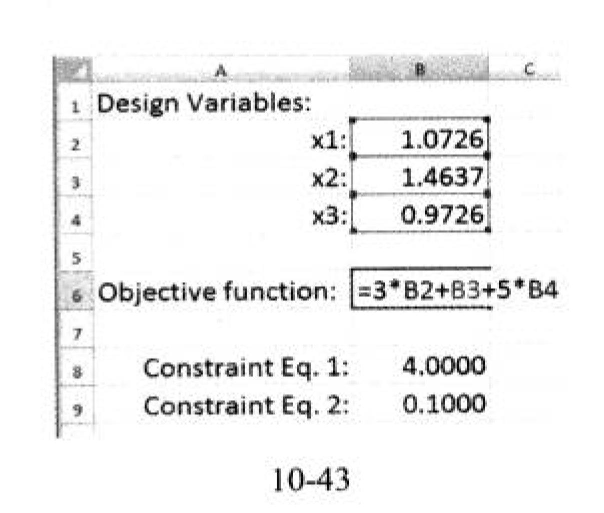
约束条件：

$$x_1 + 2x_2 \geq 4$$

$$x_1 - x_3 \leq 0.1 \tag{10-29}$$

解：

这个问题的约束条件与例 10.7 相同，但是在这个例子里，我们改变了目标函数，使它符合满足线性约束最优化问题的定义。因此我们不仅要修改电子表格，而且为了要满足这个问题的线性要求，还要修改求解方法。



10-43

首先，我们修改电子表格，在 B6 单元里插入新的目标函数，如图 10-43 所示。

按 Enter 键，启动【数据】选项卡的规划求解工具。【规划求解参数】对话框里的全部参数与前面例 10.7 相同。在第 10.2 节曾提到，线性约束问题要用到专门的求解算法，这个求解算法就是单纯形算法。该算法假定最优解出现在可行区域的某个角点位置。此外还需要设置一个选项，表示它是一个线性问题，确保采用合适的求解算法。为此，我们单击【选项】按钮，如图 10-44 所示。

出现【规划求解选项】对话框，选择【采用线性模型】选项，如图 10-45 所示。注意，在这个选项对话框里除了此线性模型外，还有其他几个选项。在复杂的最优化问题里，需要设置这些选项，在这个问题里，我们暂且选择它们的默认值。

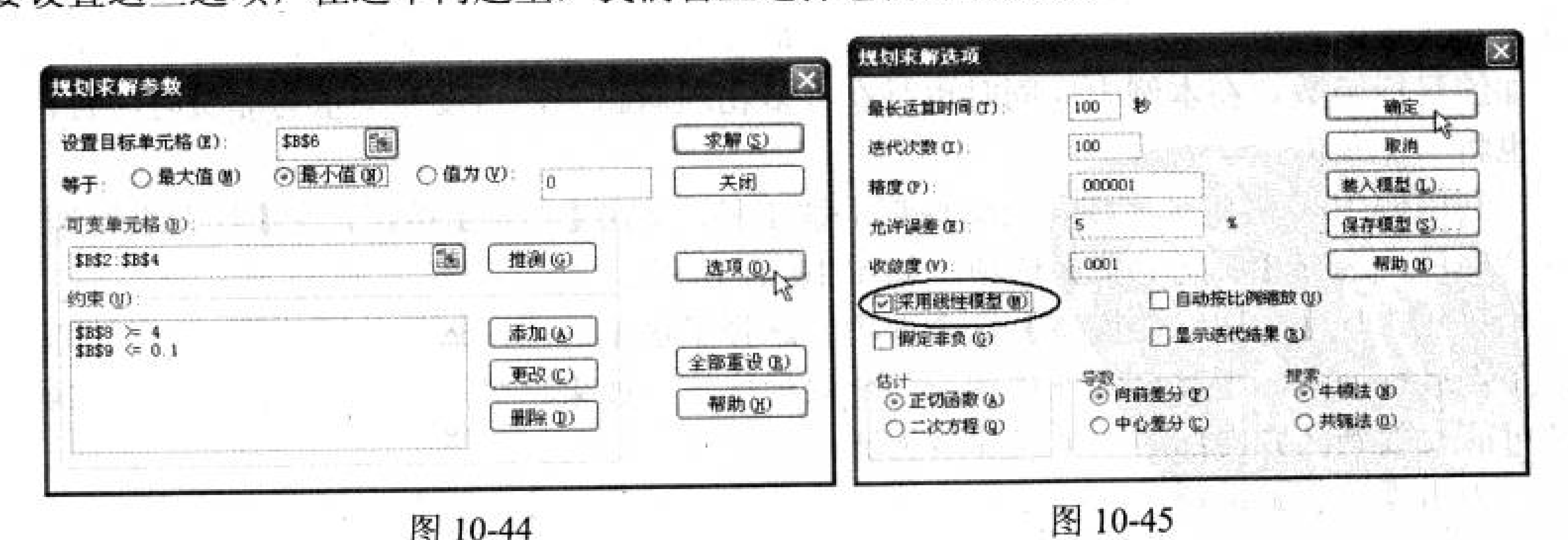


图 10-44　　图 10-45

单击【确定】按钮，关闭选项对话框，再单击【求解】按钮，执行求解算法。算法执

行结束后，出现如图 10-46 所示的窗口显示运算结果。

这个窗口显示的信息表明，算法不能收敛。这是指，这个问题的可行解区域没有边界。

单击【确定】按钮，设计变量将变为新值，但是解没有意义。

图 10-46

例 10.9

求线性约束函数的最小值(续)。

求解下面的最优化问题。

最小值：

$$y = 3x_1 + x_2 + 5x_3$$

约束条件：

$$x_1 + 2x_2 \geq 4$$

$$x_1 - x_3 \leq 0.1$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_3 \geq 0 \tag{10-30}$$

解：

这个问题几乎与例 10.8 完全相同，但是为了保证设计变量为非负值，这里增加了 3 个约束条件。

为了求解这个问题，我们需要增加这些约束方程。设计变量的非负限制在工程最优化问题中非常常见。这是因为设计常量通常表示物理量度量和参数(如大小、重量、电阻、作用力等)，这些量取负值没有意义。如果采用约束方程正常的求解步骤，我们可以一次添加一个约束条件。但是由于设计变量的非负约束非常普遍，因此 Excel 里有一种快捷方法把这个约束条件作用于全部设计变量。需要提醒读者，采用这种快捷方法，还会把非负约束强加给目标函数。在本例中，同时也是在许多物理问题中，当设计变量为非负时，目标函数也将为非负值。

利用例 10.8 的电子表格，启动规划求解工具，单击【选项】按钮，在【规划求解选项】对话框里，选择【假定非负】选项，如图 10-47 所示。

仍然选择【采用线性模型】这个选项。选择了这个选项，求解算法将只对可行解区域的角点进行搜索，因此会保证找到一个全局最优解。如果这个选项没有被选择，则求解算法可能收敛到区域内部的一个局部最优解。

单击【确定】按钮，设置好规划求解的选项，再单击【求解】按钮执行算法。注意，这次只得到一个解，因为我们增加了非负约束条件，给可行区域添加了边界条。结果如图 10-48 所示。

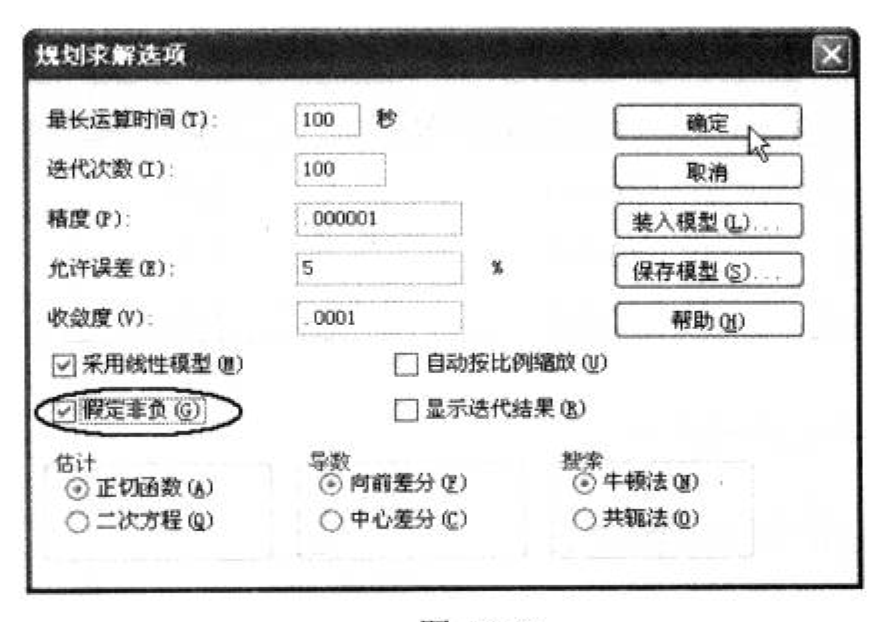


图 10-47

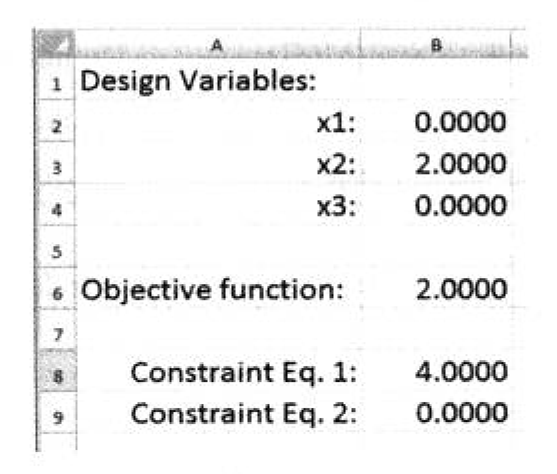


图 10-48

10.6 教程：约束线性问题在工程上的应用

当工程师要解决一个问题时，目标不仅仅是要得到这个问题的解，而是要得到一个最优解。因此，最优化处理成为工程师们最主要的任务之一。在第 10.1 节里，我们把一个典型的机械/建筑设计问题描述成一个非线性约束最优化问题。在本节中，我们将把一个典型的工业/制造工程设计问题描述成一个线性约束最优化问题，并用 Excel 求解这个问题。

例 10.10

某个化学公司在 4 个城市设立了工业粘合剂加工、储存工厂：

- 洛杉矶的新奥尔良
- 印地安那州的加里市
- 纽约州的布法罗市
- 俄勒冈州的波特兰市

每个工厂生产这种优质粘合剂每周的产量和产生成本如表 10-1 所示。

表 10-1　各粘合剂厂的产量和生产成本

工厂	产量(加仑/周)	产生成本(美元/加仑)
新奥尔良市	18 000	0.71
加里市	22 ,000	0.98
布法罗市	14 000	1.12
波特兰市	10 000	1.29

有 3 个大客户每周定期下订单，他们的地点、订购量和议价如表 10-2 所示。

表 10-2　客户的订购量、价格及地点

客户	地点	订购量(加仑/周)	价格(美元/加仑)
中西部制造公司(MMI)	Duluth,MN	26 000	2.00
西海岸公司	Stockton, CA	13 000	2.75
德州制造公司(TMC)	Houston,TX	11 000	3.00

经估算，各工厂到各客户的运费如表 10-3 所示。

表 10-3　运费

发货地\目的地	新奥尔良市(美元/加仑)	加里市(美元/加仑)	布以罗市(美元/加仑)	波特兰市(美元/加仑)
MMI	0.22	0.12	0.16	0.17
WCC	0.26	0.20	0.39	0.11
TMC	0.12	0.18	0.21	0.20

确定 4 个工厂的生产计划，使得该化学公司的利润最大化。

解：

步骤 1　确定设计变量：本例的目标就是确定每个工厂每周的产量多少，卖给每个顾客多少。因此，一共有 12 个设计变量。

x_1：新奥尔良工厂生产的销售给 MMI 的数量

x_2：新奥尔良工厂生产的销售给 WCC 的数量

x_3：新奥尔良工厂生产的销售给 TMC 的数量

x_4：加里工厂生产的销售给 MMI 的数量

x_5：加里工厂生产的销售给 WCC 的数量

x_6：加里工厂生产的销售给 TMC 的数量

x_7：法布罗工厂产生的销售给 MMI 的数量

x_8：法布罗工厂产生的销售给 WCC 的数量

x_9：法布罗工厂产生的销售给 TMC 的数量

x_{10}：波特兰工厂生产的销售给 MMI 的数量

x_{11}：波特兰工厂生产的销售给 WCC 的数量

x_{12}：波特兰工厂生产的销售给 TMC 的数量

步骤 2：建立目标函数。该公司的生产计划的目的是使利润最大化。考虑到每加仑的销售价格、生产成本和运费。每加仑利润的计算公式如下：

最大值：

$$P=(2.00-0.71-0.22)x_1+(2.75-0.71-0.26)x_2+(3.00-0.71-0.12)x_3$$
$$+(2.00-0.98-0.12)x_4+(2.75-0.98-0.20)x_5+(3.00-0.98-0.18)x_6$$

$$+ (2.00 - 1.12 - 0.16)x_7 + (2.75 - 1.12 - .39)x_8 + (3.00 - 1.12 - 0.21)x_9$$

$$+ (2.00 - 1.29 - 0.17)x_{10} + (2.75 - 1.29 - 0.11)x_{11} + (3.00 - 1.29 - 0.20)x_{12} \quad (10\text{-}31)$$

或者化简得到最大值

$$P = 1.07x_1 + 1.78x_2 + 2.17x_3 + 0.90x_4 + 1.57x_5 + 1.84x_6$$

$$+ 0.72x_7 + 1.24x_8 + 1.67x_9 + 0.54x_{10} + 1.35x_{11} + 1.51x_{12} \quad (10\text{-}32)$$

步骤 3：确定约束条件。每个工厂受产量限制，因此有四个约束条件限制了该公司的生产能力。例如，新奥尔良工厂最多只能生产 18 000gul，我们把它表示成约束方程如下：

$$x_1 + x_2 + x_3 \le 18\,000 \quad (10\text{-}33)$$

同样道理，我们给出其他 3 个约束条件：

$$x_4 + x_5 + x_6 \le 22\,000 \quad (10\text{-}34)$$

$$x_7 + x_8 + x_9 \le 14\,000 \quad (10\text{-}35)$$

$$x_{10} + x_{11} + x_{12} \le 10\,000 \quad (10\text{-}36)$$

此外，必须满足每个客户的需求，例如，MMI 公司的订购量是 26 000，则把它表示为约束条件：

$$x_1 + x_4 + x_7 + x_{10} = 26\,000 \quad (10\text{-}37)$$

同样道理，我们得到另外两个约束条件：

$$x_2 + x_5 + x_8 + x_{11} = 13\,000 \quad (10\text{-}38)$$

$$x_3 + x_6 + x_9 + x_{12} = 11\,000 \quad (10\text{-}39)$$

其他约束条件隐含在这些变量的非负特性里(本例中，它们是产量)，因此有：

$$x_1, \ldots, x_{12} \ge 0 \quad (10\text{-}40)$$

由于这个问题属于多变量约束问题，我们选择 Excel 的规划求解工具。需要在已建立电子表格中，输入目标函数公式和约束条件公式。我们按图 10-49 的格式，在这个电子表格里输入公式，并且设置格式。把初始值设置为 0。

	A	B	C	E	F
1	***Design Variables:***			***Objective Function:***	
2	New Orleans to MMI	x1=	0		Profit = =1.07*C2+1.78*C3+2.17*C4+0.9*C5+1.57*C6+1.84*C7+0.72*C8+1.24*C9+1.67*C10+0.54*C11+1.35*C12+1.51*C13
3	New Orleans to WCC	x2=	0		
4	New Orleans to TMC	x3=	0		
5	Gary to MMI	x4=	0	***Constraints:***	
6	Gary to WCC	x5=	0	Capacity for New Orleans:	=C2+C3+C4
7	Gary to TMC	x6=	0	Capacity for Gary:	=C5+C6+C7
8	Buffalo to MMI	x7=	0	Capacity for Buffalo:	=C8+C9+C10
9	Buffalo to WCC	x8=	0	Capacity for Portland:	=C11+C12+C13
10	Buffalo to TMC	x9=	0	Demand for MMI:	=C2+C5+C8+C11
11	Portland to MMI	x10=	0	Demand for WCC:	=C3+C6+C9+C12
12	Portland to WCC	x11=	0	Demand for TMC:	=C4+C7+C10+C13
13	Portland to TMC	x12=	0		
14					

图 10-49

启动规划求解程序，并按图 10-50 的格式，设置好参数。

注意，虽然图 10-50 只显示了前面 6 个约束方程，但实际上我们要输入全部 7 个约束方程。

由于这是一个线性问题，同时设计变量必须是非负的，因此必须把规划求解工具的选项设置为如图 10-51 的格式。

规划求解参数

设置目标单元格(E)：　　　　求解(S)

等于：⊙最大值(M)　○最小值(N)　○值为(V)：0　　　　关闭

可变单元格(B)：

$C$2:$C$13　　推测(G)　　选项(O)

约束(U)：

$F$10 = 26000
$F$11 = 13000
$F$12 = 11000
$F$6 <= 18000
$F$7 <= 22000
$F$8 <= 14000
$F$9 <= 11000

添加(A)　更改(C)　删除(D)　全部重设(R)　帮助(H)

图 10-50

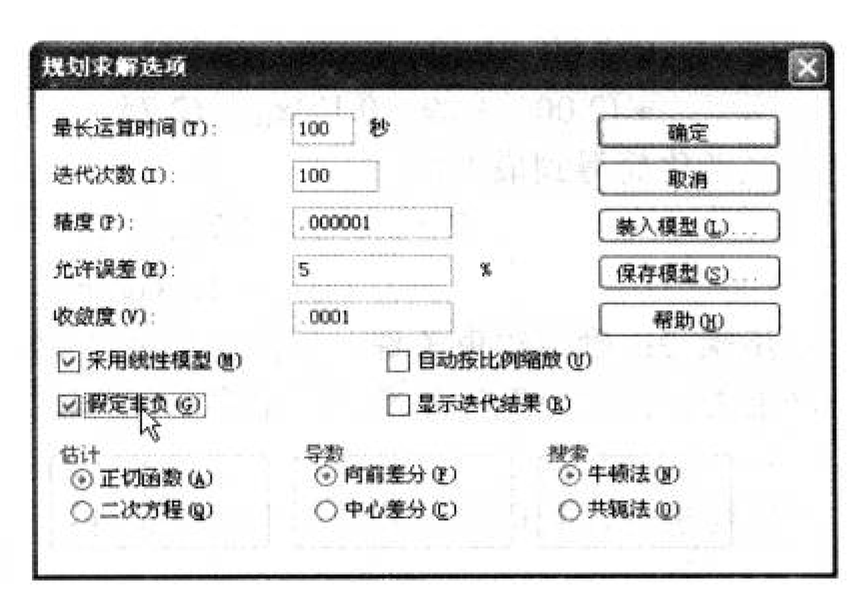


图 10-51

执行求解程序，结果如图 10-52 所示。

	A	B	C	D	E	F
1	***Design Variables:***				***Objective Function:***	
2	New Orleans to MMI	x1=	0		Profit =	67350
3	New Orleans to WCC	x2=	7000			
4	New Orleans to TMC	x3=	11000			
5	Gary to MMI	x4=	16000		***Constraints:***	
6	Gary to WCC	x5=	6000		Capacity for New Orleans:	18000
7	Gary to TMC	x6=	0		Capacity for Gary:	22000
8	Buffalo to MMI	x7=	10000		Capacity for Buffalo:	10000
9	Buffalo to WCC	x8=	0		Capacity for Portland:	0
10	Buffalo to TMC	x9=	0		Demand for MMI:	26000
11	Portland to MMI	x10=	0		Demand for WCC:	13000
12	Portland to WCC	x11=	0		Demand for TMC:	11000
13	Portland to TMC	x12=	0			

图 10-52

该结果表明，以下的生产计划才是最优的：

- 新奥尔良工厂生产 18 000gal，其中：
  - 7 000gal 销售给西海岸公司。
  - 11 000gal 销售给德州制造公司。
- 加里工厂生产 22 000gal，其中：
  - 16000 加仑提供给中西部制造公司。
  - 6000 加仑提供给西海岸公司。
- 布法罗工厂生产 10 000gal，全部销售给中西部制造公司。

这个模型还预测出，如果公司按这种最优化产生方案进行生产，该生产线每周的利润为 67 350 美元。需要特别说明的是，波特兰工厂没有生产任务，布法罗工厂还有 4 000gal 的生产能力没有利用起来。

10.7 习题

1. 用MATLAB的fminsearch命令求函数$y = 3x^4 - 2x^2 + 4x + 1$ 的最小值。并绘制这个函数的曲线验证求得的结果。

2. 用 Excel 重做习题 1。

3. 用MATLAB的fminsearch命令求多变量函数$y = 3x^4 - 2x^2 + 4x + 1 + + z^2 - z$的最小值。

4. 用 Excel 重做习题 3。

5. 用MATLAB的fminbnd命令求函数$y = x^2 \cos(2\pi x)$ 在 $0 \leqslant x \leqslant 1$ 范围内的最小值，并且绘制这个函数的曲线验证求得的结果。

6. 用 Excel 的规划求解工具重做习题 5。

7. 用 Excel 重做第 10.1 节的横梁设计问题。

8. 一位工程师提议在第 10.1 节的横梁设计问题中使用铝材料，而不使用钢材料。用铝材料的弹性模量 E 修改这个问题，重新计算这个最优化问题。

9. 一个学生小组参加一个机器人比赛。在这个比赛里，他们的机器人从基点位置走到目标位置抓取回物体，并返回基点位置。成功取回给分数。分数有 3 档：100、50 和 25。根据物体离基点的距离给分。比赛的时间限制为 4min，只有当物体取回到基点才能计分。该小组在预练时，曾 10 次均取回物体。取回物体所需要的时间及每个物体的得分如表 10-4 所示。

表 10-4

物体序号	分数值	取回时间(s)
1	100	110
2	100	80
3	100	75
4	50	55
5	50	45
6	50	30
7	25	22

(续表)

物体序号	分数值	取回时间(s)
8	25	20
9	25	15
10	25	15

利用 Excel 的规划求解程序，计算为了在 4min 时间内，使小组的得分最高，要取回哪些物体？

提示：这个问题的自变量为每个被取回物体的数量，由于每个物体只有一个，因此自变量的值为 1，或者为 0。因此约束条件是每个自变量为小于等于 1 的非负整数。这些约束条件可以统一操作。例如，当读者在规划求解程序里添加一个约束条件时，选取包括全部自变量的单元区域，一次性输入小于等于 1 的约束条件。

10. 重新考虑习题 9 的机器人比赛最优化问题。如果比赛的时间改为 5min，小组如何改变他们的策略？

11. 重做第 10.6 节的生产计划最优化问题。波特兰的一位工程师提出了工艺改进方法，使得每加仑的生产成本下降了 0.10 美元。问该公司如何调整其最优化生产计划？

12. 重做第 10.6 节的生产计划最优化问题。西海岸公司把每周的订购量增加到 22 000gal，但是要求每加仑的价格下降 0.5 美元。问该公司如何调整其最优化生产计划？

13. 在习题 11 和习题 12 中，我们必须修改目标函数和约束条件的方程。假如该模型应用在各种情形，那么这样修改会浪费时间，也容易产生错误。修改这个问题的电子表格，使得所有的输入参数(如产量、生产成本、客户订价和运输成本)都输入在表格最上方。如图 10-53 的阴影部分所示。修改方程显示总销售收入、总成本和每个工厂与每个客户组合的利润，以及每周的总销售额和总利润。此外，添加单元，显示每个工厂的总产量、销售给每个客户的销售量和每个工厂的多余产量。把利润按工厂和按客户进行分类。修改约束条件，引用产量和订购量所在的单元。经过上述修改后，如果我们需要修改输入参数中的一个，则规划求解模型保持不变。

把修改后的模型应用于第 10.6 节的数据，验证这个电子表格的正确性。用这个电子表格计算当以下情况发生变化时每周的利润变化：

(a) 新奥尔良的产量暂时减半，即每周 9 000bl。

(b) 德州制造公司把他们的订单增加到每周 18 000bl。

(c) 在布法罗工厂将采用一个新的劳动合同条例，它把每加仑的生产成本提高到 1.18 美元。

	A	B	C	D	E	F	G	H	I
1		Capacity,	Production			Order,	Price,		
2		gallons/week	Cost, $/gallon			gallons/week	$/gallon		
3	New Orleans	18,000	0.71		MMI	26,000	2.00		
4	Gary	22,000	0.98		WWC	13,000	2.75		
5	Buffalo	14,000	1.12		TMC	11,000	3.00		
6	Portland	10,000	1.29						
7									
8	Shipping Cost, $/gallon								
9		New Orleans	Gary	Buffalo	Portland				
10	MMI	0.22	0.12	0.16	0.17				
11	WWC	0.26	0.20	0.39	0.11				
12	TMC	0.12	0.18	0.21	0.20				
13									
14									
15			Price,	Prod. Cost,	Ship. Cost,	Volume,	Sales,	Cost,	Profit,
16	Plant	Customer	$/gallon	$/gallon	$/gallon	gallons	$	$	$
17	New Orleans	MMI	2.00	0.71	0.22	0	0	0	0
18	New Orleans	WCC	2.75	0.71	0.26	7,000	19,250	6,790	12,460
19	New Orleans	TMC	3.00	0.71	0.12	11,000	33,000	9,130	23,870
20	Gary	MMI	2.00	0.98	0.12	16,000	32,000	17,600	14,400
21	Gary	WCC	2.75	0.98	0.20	6,000	16,500	7,080	9,420
22	Gary	TMC	3.00	0.98	0.18	0	0	0	0
23	Buffalo	MMI	2.00	1.12	0.16	10,000	20,000	12,800	7,200
24	Buffalo	WCC	2.75	1.12	0.39	0	0	0	0
25	Buffalo	TMC	3.00	1.12	0.21	0	0	0	0
26	Portland	MMI	2.00	1.29	0.17	0	0	0	0
27	Portland	WCC	2.75	1.29	0.11	0	0	0	0
28	Portland	TMC	3.00	1.29	0.20	0	0	0	0
29						TOTALS	120,750	53,400	67,350
30									
31	TOTAL PRODUCTION, gallons			EXCESS CAPACITY, gallons			PROFIT BY CUSTOMER, $		
32	New Orleans	18,000		New Orleans	0		MMI	21,600	
33	Gary	22,000		Gary	0		WWC	21,880	
34	Buffalo	10,000		Buffalo	4,000		TMC	23,870	
35	Portland	0		Portland	10,000				
36							PROFIT BY PLANT, $		
37	TOTAL SALES VOLUMES, gallons						New Orleans	36,330	
38	MMI	26,000					Gary	23,820	
39	WWC	13,000					Buffalo	7,200	
40	TMC	11,000					Portland	0	

图 10-53

14. 数学系教授 Timothy Pennings 在他的论文 *Do Dogs Know Calculus*？(狗了懂得微积分吗)中描述了在他与他的宠物小狗 Welsh Corgi,Elvis 玩传球游戏时观察到的一件非常有趣的事。他站在密歇根湖的岸上(图 10-54 中的 A 点)把球投到湖里(图 10-54 中 B 点)。Elvis 既不直接从 A 点游泳到 B 点(路径 a)，也不先沿着湖岸跑，跑到离 B 最近点再游泳过去(路径 b)，而是选择一个最优化的路径(路径 c)，如果考虑到跑的速度比游泳的速度快这个因素，Elvis 按路径 c 取回球所花的时间最少。

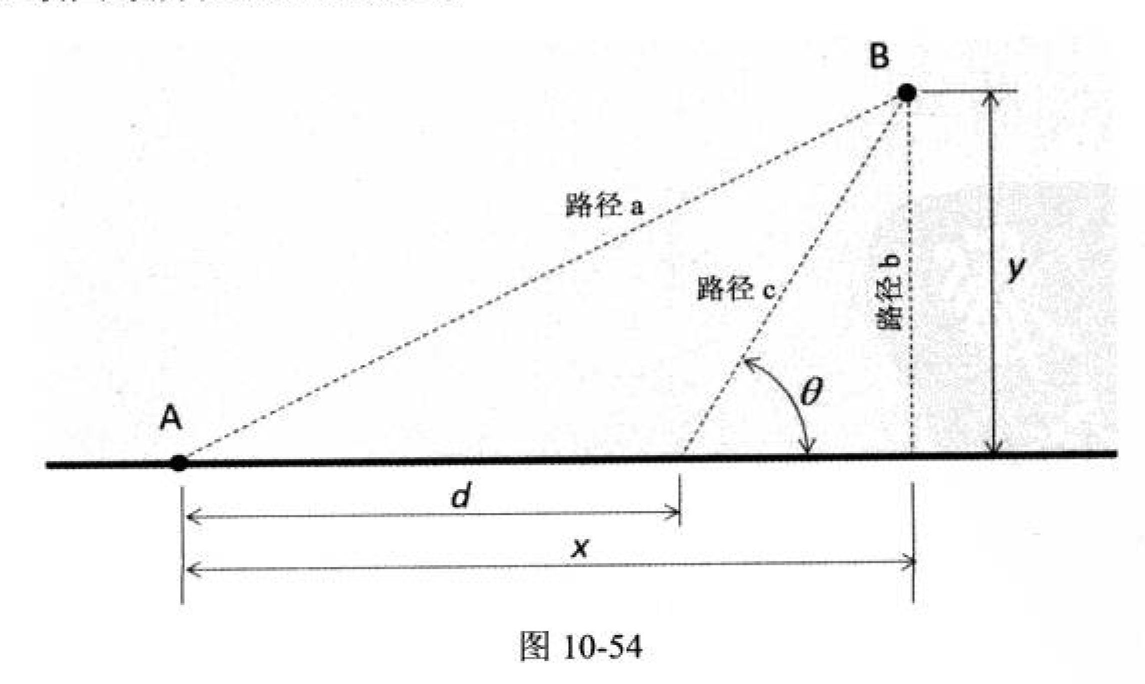


图 10-54

(a) 用 Excel 的规划求解程序，Elvis 沿湖边跑多少距离(即 d)，才会使取回球所花的时间最小，假设：

- Elvis 的跑步速度为 6.4m/s。
- Elvis 的游泳速度为 0.91m/s。
- x=30.0m。
- y=10.9m

同时求路径 c 相对于湖岸的角度 θ 为多大？

(b) 假设 x=20m，y=15m，重新计算这个问题。

说明：Pennings 教授根据他的解析解，发现角度 θ 与跑步与游泳的相对速度有关。因此它与 x 和 y 无关。他还证明，Elvis 选择路径与最优化路径相符合。